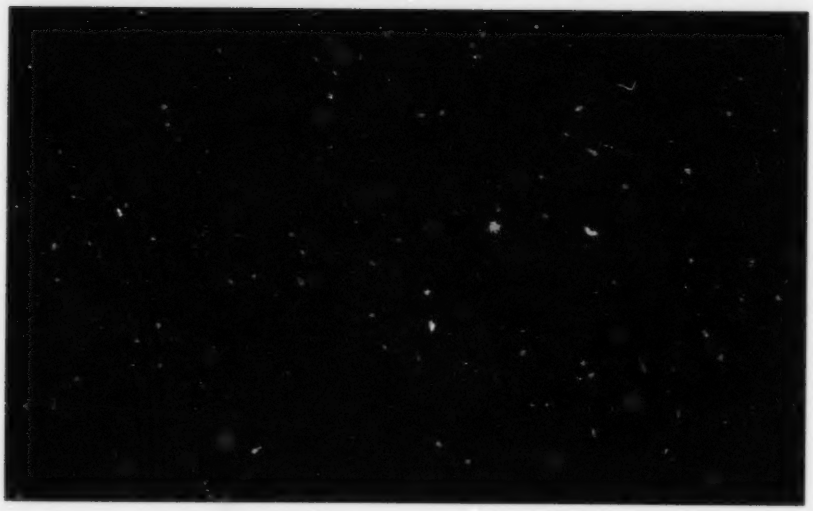
CENTRE DE RECHERCHE
SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE



TRAVAUX D'ÉTUDIANT

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A,
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: 05-24146 BIBPOLYTEC
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

91-02549/

CRT-760

L'algorithme de Karmarkar, ses variantes et ses extensions*

par

Jean-François Mondou**

* Travail effectué dans le cadre de la seconde partie de l'examen de synthèse.

** Étudiant de Ph.D., département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, succursale A, Montréal, Canada H3C 3J7.

Centre de recherche sur les transports — Publication #760

Février 1991

Dépôt légal – 1er trimestre 1991

RESUME

La publication par Khachian en 1979 d'une preuve établissant que le nombre d'opérations élémentaires requis par la méthode des ellipsoïdes pour solutionner un programme linéaire est borné par un polynôme a soulevé peu d'enthousiasme chez les chercheurs opérationnelles. Bien que très riche en résultats théoriques, cette procédure est inefficace en pratique. Or, depuis 1984, l'algorithme mis au point par Karmarkar ne cesse de soulever l'intérêt des théoriciens et des praticiens de la recherche opérationnelle. Pour compte, à ce moment là il a été proclamé que cet algorithme est cinquante fois plus rapide que les meilleures implantations de la méthode du simplexe!

Le but de ce travail est de donner une vue d'ensemble des méthodes de points intérieurs qui sont apparues depuis la publication des travaux de Karmarkar. Celles-ci sont classifiées en deux groupes. Le premier groupe est constitué des extensions et des variantes de l'algorithme de Karmarkar. Parmi ces algorithmes, nous distinguons ceux qui ont recours aux *transformations projectives* et ceux qui emploient des *transformations affines*. Les *méthodes de trajectoire*, issues des résultats classiques de l'étude des programmes non linéaires, forment la seconde catégorie.

Cependant, afin d'éviter une litanie inutile d'algorithmes, nous avons préféré mettre l'emphase sur les motivations et les difficultés sous-jacentes à ces méthodes, sans toutefois recourir à un formalisme mathématique trop lourd. A cet effet, nous avons trouvé dans la littérature une méthode générale (très souvent ignorée) dont l'analyse permet une synthèse de plusieurs de ces méthodes.

MOTS CLEFS

ALGORITHME DE KARMARKAR,

TRANSFORMATIONS PROJECTIVES,

TRANSFORMATIONS AFFINES,

METHODES DE TRAJECTOIRE.

INTRODUCTION

Il est convenu d'associer la naissance de la programmation mathématique à la mise au point de l'algorithme du simplexe en 1947. Cet algorithme effectue la résolution d'un programme linéaire par un examen des points extrêmes du polyèdre défini par les contraintes. Bien que le nombre d'itérations requis par l'algorithme du simplexe ne soit pas borné par un polynôme, son efficacité pratique est très connue. Dantzig (1963) fait toutefois ressortir le caractère insatisfaisant de la recherche d'une solution en suivant les frontières du domaine réalisable. Le débat à savoir s'il existe une procédure efficace, tant du point de vue pratique que théorique, atteignant la solution optimale par l'intérieur du domaine réalisable (appelée *méthode de points intérieurs*) était lancé.

La publication par Khachian en 1979 (des notes historiques se trouvent dans Goldfarb et Todd, 1986) d'une preuve établissant que le nombre d'opérations élémentaires requis par la méthode des ellipsoïdes pour solutionner un programme linéaire est borné par un polynôme a soulevé peu d'enthousiasme chez les chercheurs opérationnelles. Bien que très riche en résultats théoriques, cette procédure est inefficace en pratique.

Or depuis 1984, l'algorithme mis au point par Karmarkar ne cesse de soulever l'intérêt des théoriciens et des praticiens de la recherche opérationnelle. Pour compte, à ce moment là il a été proclamé que cet algorithme est cinquante fois plus rapide que les meilleures implantations de la méthode du simplexe! De plus, considérant les implications financières très importantes (le système KORBX mis au point par les laboratoires AT&T pour résoudre des programmes linéaires de grandes tailles coûte environ neuf million de dollars), et l'attitude des laboratoires AT&T face à la communauté scientifique (voir Hooker, 1986), une certaine forme de compétition a pris naissance (voir Marsten et al., 1990). (Cette compétition a d'ailleurs révélée que l'estimation initiale de la performance de l'algorithme de Karmarkar était très optimiste!) C'est donc dire que les méthodes de points intérieurs ne sont plus tabous (certains niveaux d'aspiration ont été dépassés!).

Les méthodes de points intérieurs sont classifiées en deux groupes. Le premier groupe est constitué des extensions et des variantes de l'algorithme de Karmarkar. Parmi ces algorithmes, nous distinguons ceux qui ont recours aux *transformations projectives* et ceux qui emploient des *transformations affines*. Les *méthodes de trajectoire*, issues des résultats classiques de l'étude des programmes non linéaires, forment la seconde catégorie.

Une analyse sérieuse des algorithmes de points intérieurs est difficile à cause du nombre élevé de publications et des connaissances très poussées, tant en mathématique qu'en informatique, requises pour les maîtriser. Afin d'éviter une litanie inutile d'algorithmes, nous avons préféré montrer au lecteur les motivations et les difficultés sous-jacentes à ces méthodes, sans toutefois recourir à un formalisme mathématique trop lourd. A cet effet nous avons trouvé dans la littérature une méthode générale dont l'analyse permet une synthèse de l'algorithme de Karmarkar et ses variantes. Le but fixé est de donner une vue d'ensemble des méthodes de points intérieurs afin d'en relever les aspects importants. Ce travail est divisé en cinq sections.

Tout d'abord, nous résumons l'algorithme de Karmarkar qui sert à résoudre un programme linéaire dont le système des contraintes est l'intersection de l'espace nul d'une matrice avec un simplexe unité et satisfaisant les deux hypothèses: *i*) le vecteur constitué de uns est un point du domaine réalisable; *ii*) la valeur de la solution optimale est connue. Dans cette première section, nous ne tentons pas d'en dégager les fondements car l'approche de Karmarkar n'est pas intuitive.

A la place nous nous contentons d'expliquer d'une façon algébrique les ingrédients qui composent cet algorithme. Cela permet de décrire la structure particulière des contraintes et les opérations de projection afin d'en soulever les implications.

Les sections deux et trois sont réservées à l'étude détaillée de l'algorithme de Karmarkar et ses variantes. Dans ces sections nous tentons d'en comprendre l'efficacité théorique et le rôle joué par la structure particulière des contraintes qu'elles requièrent. A cet effet, nous croyons qu'il est préférable, même si cela n'est pas la façon usuelle de le faire, de considérer l'algorithme de Karmarkar comme étant un algorithme de décision. De ce point de vue, la procédure permet d'affirmer ou d'infirmer l'existence d'un point appartenant à une certaine région. Ainsi la seconde section contient un rappel de quelques résultats issus de la théorie de la complexité. Ceux-ci sont appliqués à l'étude des programmes linéaires. La troisième section est consacrée à l'étude de la méthode générale de points intérieurs pour laquelle l'algorithme de Karmarkar est un cas particulier.

Dans la quatrième section nous explicitons les travaux de plusieurs chercheurs qui ont mis au point des procédures usant de transformations affines au lieu des transformations projectives. Ces algorithmes sont plus simples car ils s'apparentent à la méthode bien connue du gradient projeté. Du point de vue pratique, l'avantage principal est le traitement direct des programmes linéaires.

Enfin, suite aux travaux de Gill et al. (1986) établissant une équivalence entre l'approche de Karmarkar et les méthodes classiques issues de l'étude des programmes non linéaires (Lagrange, Newton, Fiacco et McCormick), plusieurs auteurs les ont spécialisés avec succès pour la résolution des programmes linéaires. Ce sont les méthodes de trajectoire et nous les décrivons dans la cinquième section.

Nous concluons ce travail en soulignant les aspects numériques et pratiques des algorithmes de points intérieurs. Nous discutons aussi des impacts de ces nouvelles méthodes sur la programmation mathématique en général.

Nous employons les notations standards adoptées dans la littérature de la programmation mathématique (voir par exemple Bazarra et al. 1990, Bazarra et Jarvis, 1979). En particulier, les vecteurs sont représentés par des lettres minuscules imprimées en gras tandis que les lettres majuscules imprimées en gras sont utilisées pour les matrices. Tous les vecteurs sont des vecteurs colonnes; le vecteur $(1, 1, \ldots, 1)^T$ est noté $\mathbf{1}$ et $\mathbf{0} = (0, 0, \ldots, 0)^T$. L'espace euclidien de dimension n est $\mathbf{E}^n$, $\mathbf{E}^n_+$ est l'octant positif et $\text{int}\mathbf{E}^n_+$ est son intérieur. Enfin, "log" désigne le logarithme pris en base deux tandis que "ln" est le logarithme népérien.

1) L'ALGORITHME DE KARMARKAR

Dans cette section nous décrivons la méthode originale de Karmarkar. Toutefois, la description de Karmarkar (1984) étant ardue, cette partie est inspirée des articles de Rockett et Stevenson (1987), Goldfarb et Todd (1989) et le chapitre huit du livre de Bazarra et al. (1990). De plus, nous ne tentons pas ici d'en dégager les fondements. L'algorithme de Karmarkar n'est pas intuitif à cause principalement de la structure particulière des contraintes et du rôle joué par les opérations de projection. A la place nous nous contentons d'en expliquer d'une façon algébrique les ingrédients afin d'en soulever les questions pertinentes. Cela permet de mieux saisir les différences, parfois subtiles, entre les diverses variantes et extensions de la méthode de Karmarkar. Les deux prochaines sections contiennent une analyse plus détaillée.

L'algorithme de Karmarkar a recours à des opérations de projection et de normalisation afin de résoudre un programme linéaire dont le système des contraintes est l'intersection de l'espace nul d'une matrice avec un simplexe unité et satisfaisant les deux hypothèses: *i*) le vecteur constitué de uns est un point du domaine réalisable; *ii*) la valeur de la solution optimale est connue. Pour cette raison, l'algorithme de Karmarkar est appelé algorithme de projection (voir Hooker, 1986) et le programme linéaire qu'il résout est appelé programme canonique de Karmarkar:

$$\text{minimiser } \{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0},\ \mathbf{x} \in \Delta\} \tag{1.1}$$

où $\mathbf{c}$ est le vecteur de dimension n des coefficients de la fonction à minimiser, $\mathbf{A}$ est une matrice $m \times n$ des coefficients des contraintes et $\mathbf{x} \in \mathbf{E}^n$ est le vecteur des inconnues. L'analyse de la complexité théorique requiert aussi que tous les coefficients soient des nombres entiers. Enfin Δ est le simplexe unité contenu dans $\mathbf{E}^n$, c'est-à-dire $\Delta = \{\mathbf{z} \in \mathbf{E}^n : \mathbf{1}^T\mathbf{z} = 1,\ \mathbf{z} \geq \mathbf{0}\}$. Le programme (1.1) doit aussi assumer les hypothèses suivantes:

(H1) $\mathbf{A}$ est une matrice de rang plein;

(H2) La valeur optimale de ce programme linéaire est zéro; et

(H3) L'intérieur du domaine réalisable n'est pas vide.

A première vue, le programme linéaire (1.1) et les hypothèses qui l'accompagnent semblent très restrictives. Une approche pour contourner l'hypothèse (H2) est mentionnée dans la seconde section tandis que la section 3 contient plusieurs manipulations pour mettre un programme linéaire général sous la forme (1.1). D'autres façons de traiter l'hypothèse (H2) sont aussi mentionnées. Noter toutefois que la première hypothèse est standard à l'étude des programmes linéaires tandis que la seconde stipule que la valeur de la solution optimale soit connue. En effet, si z^* est la valeur de la solution optimale de (1.1), alors remplacer $\mathbf{c}$ par $(\mathbf{c} - z^*\mathbf{1})$ dans la fonction objectif donne lieu à un programme linéaire équivalent ayant zéro comme plus petite valeur. Enfin, la troisième hypothèse stipule qu'il existe un vecteur $\bar{\mathbf{x}}$ vérifiant: $\mathbf{A}\bar{\mathbf{x}} = \mathbf{0},\ \mathbf{1}^T\bar{\mathbf{x}} = 1,\ \bar{\mathbf{x}} \in \text{int}\mathbf{E}^n_+$. Par la suite un tel point est appelé point intérieur du domaine réalisable.

Le domaine réalisable de (1.1) est l'intersection d'un sous-espace linéaire de dimension $n - m$ (qui est défini par le système d'équations homogènes $\mathbf{A}\mathbf{x} = \mathbf{0}$; c'est l'espace nul de la matrice $\mathbf{A}$) avec le simplexe unité Δ de dimension $n - 1$. Le domaine réalisable est donc une région de dimension $n - m - 1$ qui est un polyèdre convexe. Noter que $\{\mathbf{A}\mathbf{x} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}\}$ est un cône issu de l'origine,

l'espace nul de la matrice $\mathbf{A}$ est donc constitué de rayons (au lieu de vecteurs) et dans ce cas le rôle du simplexe unité est une contrainte de normalisation. Le centre de ce simplexe est le vecteur $\mathbf{1}/n$. Les ingrédients employés par l'algorithme de Karmarkar sont: une opération de projection, une boule dans un espace affine, un pas de la méthode du gradient projeté.

En premier lieu, définissons la matrice diagonale $\overline{\mathbf{X}}$ formée des composantes d'un point intérieur $\overline{\mathbf{x}}$ quelconque de (1.1) telle que $\overline{\mathbf{X}}\mathbf{1} = \overline{\mathbf{x}}$ et effectuons le changement de variables suivant:

$$\mathbf{y} = \frac{\overline{\mathbf{X}}^{-1}\mathbf{x}}{\mathbf{1}^T\overline{\mathbf{X}}^{-1}\mathbf{x}} \qquad \text{c'est-à-dire} \qquad y_j = \frac{x_j/\overline{x}_j}{\sum_{j=1}^{n} x_j/\overline{x}_j} \quad j = 1,2,\ldots,n \tag{1.2}$$

Ce changement de variables admet la transformation inverse unique:

$$\mathbf{x} = \frac{\overline{\mathbf{X}}\mathbf{y}}{\mathbf{1}^T\overline{\mathbf{X}}\mathbf{y}} \qquad \text{c'est-à-dire} \qquad x_j = \frac{\overline{x}_j y_j}{\sum_{j=1}^{n} \overline{x}_j y_j} \quad j = 1,2,\ldots,n \tag{1.3}$$

Les points extrêmes du simplexe unité ne sont pas modifiés par cette opération. De plus, dans l'espace des variables $\mathbf{y}$, le point $\overline{\mathbf{x}}$ est transporté au centre du simplexe unité Δ. Cette acrobatie de géométrie projective envoie Δ sur lui-même et le point intérieur $\overline{\mathbf{x}}$ au centre de ce simplexe, d'où le nom algorithme de projection (voir Hooker, 1986). Suite à cette opération, le programme linéaire (1.1) devient:

$$\text{minimiser} \left\{ \frac{\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y}}{\mathbf{1}^T\overline{\mathbf{X}}\mathbf{y}} \;:\; \mathbf{A}\overline{\mathbf{X}}\mathbf{y} = \mathbf{0},\ \mathbf{y} \in \Delta \right\} \tag{1.4}$$

Le domaine réalisable de (1.4) possède la même structure que le domaine réalisable de (1.1) sauf que la fonction à minimiser est non linéaire: c'est le quotient de deux fonctions linéaires. Toutefois, l'hypothèse (H2) stipule que la valeur optimale doit être zéro. Le dénominateur peut donc être ignoré (le dénominateur est positif pour tous les points intérieurs du domaine réalisable) et cela donne lieu à un programme linéaire ayant une forme identique à celle de (1.1) avec le point $\overline{\mathbf{x}}$ au centre du simplexe unité:

$$\text{minimiser} \ \{\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y} \;:\; \mathbf{A}\overline{\mathbf{X}}\mathbf{y} = \mathbf{0},\ \mathbf{y} \in \Delta\} \tag{1.5}$$

A partir du centre du simplexe unité dans l'espace transformé, un pas dans la direction de la plus forte descente est pris de façon à demeurer dans l'intérieur du domaine. Il faut éviter à tout prix d'aboutir à une frontière! Une manière élégante de le faire consiste à restreindre le domaine réalisable de (1.5) en inscrivant une boule dans le simplexe unité:

$$B(\mathbf{1}/n, \rho) \;=\; \{\mathbf{z} \in \mathbf{E}^n \;:\; \mathbf{1}^T\mathbf{z} = 1,\ |\mathbf{z} - \mathbf{1}/n| \le \rho\} \tag{1.6}$$

Cette boule appartient à l'espace affine $\{\mathbf{z} \in \mathbf{E}^n \;:\; \mathbf{1}^T\mathbf{z} = 1\}$, son rayon est ρ et son centre est le vecteur $\mathbf{1}/n$. Les relations d'inclusion mathématique suivantes sont vérifiées:

$$B(\mathbf{1}/n, \sqrt{1/n(n-1)}) \subseteq \Delta \subseteq B(\mathbf{1}/n, \sqrt{(n-1)/n})$$

Enfin, pour un $\rho > 0$, la solution optimale du programme mathématique

$$\text{minimiser } \{\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y} : \mathbf{A}\overline{\mathbf{X}}\mathbf{y} = \mathbf{0},\ \mathbf{y} \in B(\mathbf{1}/n, \rho)\} \tag{1.7}$$

est obtenue en prenant le pas ρ dans la direction réalisable de la plus forte descente:

$$\mathbf{y}(\rho) = \mathbf{1}/n - \rho\frac{\mathbf{d}}{\|\mathbf{d}\|} \tag{1.8}$$

Remarquer que le domaine réalisable de (1.7) est l'intersection du sous-espace affine $\{\mathbf{z} \in \mathbf{E}^n:\ \mathbf{Az} = \mathbf{0},\ \mathbf{1}^T\mathbf{z} = 1\}$ avec la boule $B(\mathbf{1}/n, \rho)$. La direction $\mathbf{d}$ de la plus forte pente qui est réalisable est la projection du gradient de la fonction objectif sur l'espace nul de la matrice $\mathbf{B} = \begin{bmatrix} \mathbf{A}\overline{\mathbf{X}} \\ \mathbf{1}^T \end{bmatrix}$ (pour un rappel de la méthode du gradient projeté, voir Bazarra et Jarvis, 1979):

$$\mathbf{d} = -(\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B})\overline{\mathbf{X}}\mathbf{c} \tag{1.9}$$

En pratique, le calcul de la matrice de projection n'est pas effectué. A la place, le système d'équations linéaires carré (symétrique) $\mathbf{B}\mathbf{B}^T\mathbf{w} = \mathbf{B}\overline{\mathbf{X}}\mathbf{c}$ est résolu. Ainsi la formule pour la direction de recherche est $\mathbf{d} = -\overline{\mathbf{X}}\mathbf{c} + \mathbf{B}^T\mathbf{w}$. De plus, la forme particulière de la matrice $\mathbf{B}$ permet de déterminer le vecteur $\mathbf{w}^T = (\mathbf{u}^T, v)$ en solutionnant les équations normales: (noter que $\mathbf{A}\overline{\mathbf{X}}\mathbf{1} = \mathbf{0}$ car $\overline{\mathbf{X}}\mathbf{1} = \overline{\mathbf{x}}$ et $\mathbf{A}\overline{\mathbf{x}} = \mathbf{0}$)

$$(\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)\mathbf{u} = \mathbf{A}\overline{\mathbf{X}}^2\mathbf{c} \tag{1.10}$$

$$(\mathbf{1}^T\mathbf{1})v = \mathbf{c}^T\overline{\mathbf{x}} \tag{1.11}$$

Ainsi $v = \mathbf{c}^T\overline{\mathbf{x}}/n$ et par suite l'équation de la direction de recherche (dans l'espace transformé) est la suivante:

$$\mathbf{d} = -\overline{\mathbf{X}}(\mathbf{c} - \mathbf{A}^T\mathbf{u}) + (\mathbf{c}^T\overline{\mathbf{x}}/n)\mathbf{1} \tag{1.12}$$

Dans l'espace original le vecteur $\mathbf{y}(\rho)$ correspond au vecteur:

$$\mathbf{x}(\rho) = \frac{\overline{\mathbf{X}}\mathbf{y}(\rho)}{\mathbf{1}^T\overline{\mathbf{X}}\mathbf{y}(\rho)}$$

Ayant tous les ingrédients en main, nous donnons maintenant la description formelle de l'algorithme de projection de Karmarkar pour résoudre un programme linéaire ayant la forme (1.1) et satisfaisant les hypothèses (H1)-(H3).

Algorithme 1 (algorithme de projection de Karmarkar)

L'initialisation consiste à déterminer un paramètre entier q qui sert à contrôler la précision numérique (voir section 2), un nombre réel $\alpha, 0 < \alpha < 1$. Aussi le compteur d'itérations k est mis à zéro et le point de départ est $\mathbf{x}^0 = \mathbf{1}/n$.

Une itération est constituée des étapes suivantes:

- Former la matrice diagonale $\overline{\mathbf{X}}$ avec les composantes du vecteur $\mathbf{x}^k$.
- Déterminer le vecteur $\mathbf{y}(\alpha/n)$ d'après les formules (1.8), (1.10) - (1.12).
- Poser $\mathbf{x}^{k+1} = \frac{\overline{\mathbf{X}}\mathbf{y}(\alpha/n)}{\mathbf{1}^T\overline{\mathbf{X}}\mathbf{y}(\alpha/n)}$ et mettre à jour k.

La procédure termine quand $\mathbf{c}^T\mathbf{x}^k < 2^{-q}$.

L'algorithme 1 est une procédure itérative qui engendre une suite de points intérieurs $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^k$ qui converge vers la solution optimale. C'est donc une méthode de type primal. La preuve de convergence est basée sur la réduction, à chaque itération, d'une certaine fonction par au moins une quantité constante. Le progrès vers la solution optimale est mesuré à l'aide de la fonction potentielle suivante:

$$\phi(\mathbf{x}, \mathbf{c}) = n \ln \mathbf{c}^T\mathbf{x} - \sum_{j=1}^{n} \ln x_j \qquad (1.13)$$

Le théorème principal de Karmarkar stipule que la suite de points intérieurs générée par l'algorithme 1 vérifie la propriété suivante:

$$n \ln\left[\frac{\mathbf{c}^T\mathbf{x}^k}{\mathbf{c}^T\mathbf{x}^0}\right] \leq \sum_{j=1}^{n} \ln(nx_j^k) - \frac{k}{5} \qquad (1.14)$$

Nous référons le lecteur à Bazaraa et al. (1990) où à Karmarkar (1984) pour les détails de la preuve. Ici, nous nous contentons d'énoncer les idées permettant d'aboutir à ce résultat.

Tout d'abord voici deux propriétés de la fonction potentielle:

- ϕ est invariante le long des rayons du cône $C = \{\mathbf{z} \in \mathbf{E}^n : \mathbf{Az} = \mathbf{0}, \mathbf{z} \geq \mathbf{0}\}$, c'est-à-dire $\phi(\alpha\mathbf{x}, \mathbf{c}) = \phi(\mathbf{x}, \mathbf{c}), \forall \alpha > 0, \forall \mathbf{x} \in C$;
- $\phi(\mathbf{y}, \overline{\mathbf{X}}\mathbf{c}) = \phi(\mathbf{x}, \mathbf{c}) - \sum_{j=1}^{n} \overline{x}_j$ où $\mathbf{y}$ est défini par la formule (1.2) et $\overline{\mathbf{x}}$ est un point intérieur.

Puisque $\phi(\mathbf{y}, \overline{\mathbf{X}}\mathbf{c})$ est la même chose que $\phi(\mathbf{x}, \mathbf{c})$ à une constante près alors montrer que la fonction potentielle diminue dans l'espace transformé est équivalent à montrer que la fonction potentielle diminue dans l'espace original. D'une façon intuitive si $n \log \mathbf{c}^T\mathbf{x}$ (d'une manière équivalente $\mathbf{c}^T\mathbf{x}$) ne peut être diminué alors $\sum \ln x_j$ augmente; c'est-à-dire prendre de l'altitude pour

permettre une meilleure descente. A chaque itération, le vecteur $\mathbf{y}(\rho)$ est obtenu en effectuant un pas, à partir du vecteur $\mathbf{1}/n$, dans la direction du gradient projeté. A ce nouveau point, les deux termes de la fonction potentielle satisfont les inégalités suivantes:

$$n \ln((\overline{\mathbf{X}}\mathbf{c})^T \mathbf{y}(\rho)) \leq n \ln((\overline{\mathbf{X}}\mathbf{c})^T \mathbf{1}/n) - \alpha$$

$$-\sum_{j=1}^{n} (y_j(\rho)) \leq -\sum_{j=1}^{n} (1/n) + \frac{\alpha^2}{2(1-\alpha)^2}$$

Par conséquent, nous avons

$$\phi(\mathbf{y}(\rho), \overline{\mathbf{X}}\mathbf{c}) - \phi(\mathbf{1}/n, \overline{\mathbf{X}}\mathbf{c}) \leq -\alpha + \frac{\alpha^2}{2(1-\alpha)^2} \tag{1.15}$$

Par exemple, en prenant $\alpha = \frac{1}{3}$ la fonction potentielle est réduite d'au moins $\frac{1}{5}$ unités à chaque itération.

Remarquer que la suite de points intérieurs n'atteint pas la solution optimale car celle-ci ce trouve à une frontière du domaine réalisable. Cependant, le fait que le domaine réalisable soit une polyèdre défini par l'intersection d'hyperplans assure qu'il existe un K , où K est un polynôme qui dépend des paramètres de la taille du problème, telle qu'à partir de $\mathbf{x}^K$ il est possible de déterminer une solution de base optimale. Cet aspect est relié à la précision numérique et cela est expliqué dans la prochaine section.

2) COMPLEXITE THEORIQUE DES PROGRAMMES LINEAIRES

Dans cette section nous rappelons les résultats importants qui concernent l'analyse de la complexité théorique des programmes linéaires. Nous énonçons les lemmes qui permettent de déduire l'efficacité théorique de l'algorithme de Karmarkar et nous montrons comment déterminer une solution de base. Nous terminons en indiquant une première façon de contourner l'hypothèse (H2).

Dans ce qui suit, les résultats s'appliquent à tout programme linéaire donné sous la forme standard:

$$\text{minimiser } \{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\} \tag{2.1}$$

Ici $\mathbf{A}$ est une matrice $m \times n$ et $\mathbf{x} \in \mathbf{R}^n$ est le vecteur des inconnues. Notre analyse requiert que les coefficients de la matrice $\mathbf{A}$ et des vecteurs $\mathbf{b}$ et $\mathbf{c}$ soient des nombres entiers. Ceci est sans perte de généralité car en réalité nous faisons l'hypothèse que les coefficients sont des nombres rationnels et qu'ils peuvent être rendus entiers par une opération de multiplication.

Une *instance* est un programme linéaire dont les coefficients sont fixés. La *taille* d'une instance est déterminée par le triplet (n,m,L) , où L est le nombre de bits requis pour emmagasiner les coefficients. Le nombre de bits nécessaire pour représenter l'entier i avec son signe est $\lceil 1 + \log_2(1 + |i|) \rceil$. Le nombre de bits d'une instance est donné par la formule suivante:

$$L = \lceil 1 + \log(1+m) \rceil + \lceil 1 + \log(1+n) \rceil + \sum_{j=1}^{n} [\lceil 1 + \log(1 + |c_j|) \rceil] +$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} [\lceil 1 + \log(1 + |a_{ij}|) \rceil] + \sum_{i=1}^{m} [\lceil 1 + \log(1 + |b_i|) \rceil]$$

Imaginons un algorithme pouvant solutionner en un nombre fini d'itérations un programme linéaire. La *complexité théorique dans le pire des cas* de cet algorithme est le nombre maximum d'opérations élémentaires requis pour parvenir à une solution. Cette borne est notée $T(n,m,L)$ pour mettre l'emphase sur sa dépendance avec toutes les instances possibles de taille (n,m,L) . Or, le calcul de $T(n,m,L)$ étant fastidieux et puisqu'il suffit d'en connaître le comportement asymptotique en fonction des paramètres n,m,L , la notation de Landau "O" est utilisée. Spécifier que $T(n,m,L)$ est $O(\ g(n,m,L)\ )$ signifie qu'il existe une constante positive τ telle que $T(n,m,L) \leq \tau g(n,m,L)$.

Une *base* pour le programme linéaire (2.1) est une matrice $\mathbf{B}$ formée de m colonnes linéairement indépendantes de $\mathbf{A}$. Une *solution de base* est un vecteur $\bar{\mathbf{x}}$ qui solutionne le système d'équations $\mathbf{B}\mathbf{x} = \mathbf{b}$. Ce vecteur est une *solution dc base réalisable* s'il est non négatif. Noter que toutes les solutions de base réalisables de (2.1) sont des vecteurs dont les composantes sont des nombres rationnels. L'inégalité suivante (voir le chapitre deux de Papadimitriou et Steiglitz, 1982) est très importante car elle permet de déduire les résultats présentés ci-dessous.

> Soit $\bar{\mathbf{x}}$ une solution de base de (2.1) alors
>
> $$|\bar{x}_j| \leq m!\alpha^{m-1}\beta \quad j = 1,2,\ldots,n \quad \text{où } \alpha = \max_{i,j}\{|a_{ij}|\},\ \beta = \max_{i}\{|b_i|\}\ .$$

La connaissance exacte de L n'est pas requise car c'est le comportement asymptotique du nombre maximum d'opérations élémentaires des algorithmes qui est étudié. Ainsi, une borne inférieure de L est suffisante. Papadimitriou et Steiglitz (1982) utilisent la borne suivante:

$$\underline{L} = mn + \lceil \log |P| \rceil \tag{2.2}$$

où P est le produit des coefficients non nuls des vecteurs **b** et **c** et de la matrice **A**.

Nous citons ici quelques résultats (voir chapitre huit de Papadimitriou et Steiglitz, 1982; chapitre huit de Bazaraa et al., 1990) qui établissent un lien entre la valeur numérique des solutions des programmes linéaires et la valeur de $\underline{L}$. Le premier résultat stipule que la valeur absolue de chacun des nombres rationnels qui composent une solution de base réalisable est bornée par $2^{\underline{L}}$. La norme d'une solution de base est donc plus petite ou égale à $\sqrt{n}\,2^{\underline{L}}$. La valeur optimale de (2.1) appartient à l'intervalle $(-2^{\underline{L}}, 2^{\underline{L}})$. Par suite, lorsqu'un point $\bar{\mathbf{x}}$ satisfait la condition $|\mathbf{c}^T\bar{\mathbf{x}} - z^*| < 2^{-\underline{L}}$ alors une solution de base réalisable optimale pour le programme canonique (1.1) peut être déterminée en un nombre fini d'itérations qui dépend d'un polynôme.

Nous établissons maintenant que la complexité théorique de l'algorithme de Karmarkar est borné par le polynôme $n^4\underline{L}^2$. Rappelons qu'à la fin de la section précédente nous avons établi que

$$n \ln\left[\frac{\mathbf{c}^T\mathbf{x}^k}{\mathbf{c}^T\mathbf{x}^0}\right] \leq \sum_{j=1}^{n} \ln(nx_j^k) - \frac{k}{5}\,. \tag{2.3}$$

Or,

$$\sum_{j=1}^{n} \ln(nx_j^k) = n \ln\left[n\left(\prod_{j=1}^{n} x_j^k\right)^{1/n}\right] \leq n \ln[\mathbf{1}\mathbf{x}^k] = 0 \tag{2.4}$$

car la moyenne arithmétique est toujours plus grande ou égale à la moyenne géométrique et $\mathbf{1}^T\mathbf{x}^k = 1$. Par conséquent, la suite de points intérieurs générés par l'algorithme de Karmarkar satisfait l'inégalité suivante:

$$\ln\left[\frac{\mathbf{c}^T\mathbf{x}^k}{\mathbf{c}^T\mathbf{x}^0}\right] \leq \frac{k}{5n} \quad \Leftrightarrow \quad \mathbf{c}^T\mathbf{x}^k \leq \mathbf{c}^T\mathbf{x}^0 e^{-k/5n}\,. \tag{2.5}$$

De plus, le critère d'arrêt étant $\mathbf{c}^T\mathbf{x}^k < 2^{-\underline{L}}$, l'algorithme de Karmarkar termine en au plus $k = 10n\underline{L}$ itérations. Chaque itération requiert la résolution d'un système d'équations, cela prend $O(n^3\underline{L})$ avec la méthode d'élimination de Gauss.

L'hypothèse (H2) qui accompagne le programme canonique de Karmarkar est à première vue très restrictive. Voici une manière d'employer l'algorithme 1 pour le cas où la valeur de la solution optimale n'est pas connue. Cela permet du même coup la description d'une procédure permettant de déterminer une solution de base à la fin de l'algorithme 1. Considérons le problème de décision qui est associé au programme linéaire (2.1).

Problème de décision (2.1)

Etant donnés une matrice **A**, les vecteurs **b** et **c**, et un nombre rationnel K, existe-t'il une vecteur **x** satisfaisant $\mathbf{Ax} = \mathbf{b}$, $\mathbf{c}^T\mathbf{x} \leq K$ et $\mathbf{x} \geq \mathbf{0}$?

Un algorithme résout ce problème de décision s'il produit la solution "*oui*" quand il est appliqué à une instance d'un problème de décision qui est affirmative. La solution est "*non*" lorsque l'algorithme a épuisé le nombre maximum d'opérations élémentaires sans produire la solution "*oui*".

Remarquer qu'un algorithme qui conduit à une solution du programme linéaire (2.1) peut être employé pour résoudre le problème de décision qui lui est associé. D'autre part, un programme linéaire admettant une solution optimale finie peut être solutionné en utilisant un algorithme applicable au problème de décision. En effet, une recherche dichotomique qui débute avec $K = 0$ permet de déterminer un intervalle qui contient la valeur de la solution optimale. Par exemple, si la longueur de l'intervalle est fixée à 2^{-2L}, la recherche termine avec une solution optimale de (2.1) après avoir appliqué $3^L + 2$ fois l'algorithme 1 pour affirmer ou infirmer l'existence d'un vecteur $\mathbf{x}$ satisfaisant $\mathbf{Ax} = \mathbf{0}$, $\mathbf{c}^T\mathbf{x} \leq K$ et $\mathbf{x} \in \Delta$.

Afin de déterminer une sclution de base optimale à partir du point intérieur $\mathbf{x}^k$, un rayon appartenant à l'espace nul des contraintes qui sont satisfaites à égalité est choisi. Une telle direction existe car cette matrice est de rang plein. Ensuite, faire un pas dans la direction de $\mathbf{d}$ si $\mathbf{c}^T\mathbf{d} < 0$ et dans la direction de $-\mathbf{d}$ sinon. Puisque le domaine réalisable de (1.1) est borné, ce pas conduit à une frontière où au moins une variable x_j est nulle. Cette procédure prend au plus $n - m - 1$ itérations et la solution obtenue $\mathbf{x}^*$ satisfait toujours $\mathbf{c}^T\mathbf{x}^* < 2^{-L}$. Cette condition est suffisante pour assurer que $\mathbf{x}^*$ est une solution de base optimale du programme canonique (1.1). Kojima (1986) a aussi établi une condition suffisante permettant de savoir si une variable est positive dans la solution de base optimale. Ce critère peut être incorporé dans l'algorithme 1 afin d'éliminer au fur et à mesure les variables qui sont dans la base optimale.

Noter qu'en pratique la valeur de $\underline{L}$ est très grande. A la place, un paramètre entier q qui dépend de la précision de l'ordinateur est utilisé.

La complexité théorique de l'algorithme 1 dépend donc du nombre de bits nécessaires pour emmagasiner les coefficients. Karmarkar (1984) fait d'ailleurs remarquer que la complexité théorique de tout algorithme pour résoudre des programmes linéaires doir avoir le paramètre L à cause de l'effort requis pour déterminer la direction de recherche. Cela contraste avec la complexité théorique de la plupart des algorithmes employés en recherche opérationnelle. C'est peut être pour cette raison que les articles lus pour ce travail traitent assez grossièrement cet aspect. Pour cette raison nous avons voulu, à tous le moins, citer les résultats qui permettent de déterminer avec précision la complexité de l'algorithme de Karmarkar.

3) ANALYSE DE LA METHODE DE KARMARKAR

Dans cette section nous analysons plus en détail l'algorithme de Karmarkar. Le point de vue adopté est celui des algorithmes de décision. Cette partie est inspirée d'un article (très souvent ignoré) décrivant une procédure générale pour déterminer un point appartenant à l'intersection d'un ensemble d'hyperplans, pour laquelle la méthode de Karmarkar est un cas particulier. L'auteur, Vial (1989), affirme de plus que sa méthode représente un cadre unificateur pour décrire les algorithmes de projection. L'analyse de Vial est intéressante car elle permet de mieux comprendre le rôle joué par la structure particulière des contraintes du programme canonique (1.1). Cela nous permet aussi d'énoncer quelques résultats qui motivent l'utilisation d'une fonction *potentielle* pour mesurer le progrès vers la solution optimale.

Cette section est divisée en quatre sous sections. En premier lieu nous montrons plusieurs transformations qui permettent de mettre un programme linéaire général sous une forme équivalente à la forme (1.1). Nous décrivons la méthode générale pour résoudre le problème de décision (2.1). Ensuite nous indiquons l'équivalence avec l'algorithme Karmarkar et nous terminons en pointant d'autres variantes et extensions.

3.1) Transformation des programmes linéaires — espaces de projection

Plus spécifiquement, le problème que nous traitons dans cette section est la détermination d'un vecteur appartenant au polyèdre

$$\{\mathbf{x} \in \mathbf{E}^n \ : \ \mathbf{M}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\} \tag{3.1}$$

où $\mathbf{M}$ est une matrice $m \times n$ et $\mathbf{b} \in \mathbf{E}^m$. Cela correspond, par exemple, au problème de décision (2.1) dans lequel une variable d'écart a été ajouté à la dernière contrainte. D'une façon générale (voir de Ghellinck et Vial, 1986; Gonzaga, 1989; Vial, 1989), l'étude des méthodes de projection requiert plutôt la forme suivante:

$$\{\mathbf{x} \in \mathbf{E}^q \ : \ \mathbf{A}\mathbf{x} = \mathbf{0}, \ \mathbf{g}^T\mathbf{x} = 1, \ \mathbf{x} \geq \mathbf{0}\} \tag{3.2}$$

où $\mathbf{A}$ est une matrice $p \times q$, et $\mathbf{g}$ est un vecteur de $\mathbf{E}^q_+$. Nous utilisons les lettres p et q et les mêmes variables $\mathbf{x}$ pour signifier que le passage de (3.1) à (3.2) requiert au moins une variables de plus et parfois une contrainte supplémentaire. Le système (3.2) définit un espace de projection car $\{\mathbf{z} \in \mathbf{E}^q \ : \ \mathbf{A}\mathbf{z} = \mathbf{0}\}$ est un sous espace linéaire de $\mathbf{E}^q$, c'est l'espace nul de la matrice $\mathbf{A}$, et $\mathbf{g}^T\mathbf{x}$ est une équation de normalisation. Dans certains cas la contrainte de normalisation est inutile et la procédure que nous décrivons requiert la forme:

$$\{\mathbf{x} \in \mathbf{E}^q \ : \ \mathbf{A}\mathbf{x} = \mathbf{0}, \ \mathbf{x} \geq \mathbf{0}, \ \mathbf{x} \neq \mathbf{0}\} \tag{3.3}$$

En premier lieu, nous montrons que les solutions des problèmes (3.2) et (3.3) sont équivalentes lorsque le cône $\{\mathbf{x} \in \mathbf{E}^q \ : \ \mathbf{A}\mathbf{x} = \mathbf{0}, \ \mathbf{g}^T\mathbf{x} = 0, \ \mathbf{x} \neq \mathbf{0}\}$ est uniquement constitué du vecteur nul. C'est l'hypothèse suivante (voir de Ghellinck et Vial, 1986; Gonzaga, 1989; Vial, 1989):

$$\{\mathbf{x} \in \mathbf{E}^q \ : \ \mathbf{A}\mathbf{x} = \mathbf{0}, \ \mathbf{g}^T\mathbf{x} = 0, \ \mathbf{x} \geq \mathbf{0}\} \quad \Rightarrow \quad \mathbf{x} = \mathbf{0} \tag{3.4}$$

Lemme 3.1

Soit un vecteur $\mathbf{g}$ de $\mathbf{E}_+^n$. Les systèmes d'équations et d'inéquations (3.2) et (3.3) sont équivalents lorsque l'hypothèse (3.4) est satisfaite.

Preuve

La condition est nécessaire car une solution $\bar{\mathbf{x}}$ de (3.2) est aussi une solution de (3.3). Noter que $\mathbf{g}^T\bar{\mathbf{x}} = 1$ implique $\bar{\mathbf{x}} \neq \mathbf{0}$.

La condition est aussi suffisante car l'existence d'une solution $\bar{\mathbf{x}}$ de (3.3) assure l'existence d'une solution de (3.2) quand (3.4) est vérifié. En effet, si $\mathbf{g}^T\bar{\mathbf{x}} = 0$ alors $\bar{\mathbf{x}} = \mathbf{0}$, contredisant le fait que $\bar{\mathbf{x}}$ soit une solution de (3.3). Nous avons donc $\mathbf{g}^T\bar{\mathbf{x}} \neq 0$, $\mathbf{g} \geq \mathbf{0}$ et $\bar{\mathbf{x}} \neq \mathbf{0}$, par suite $\mathbf{g}^T\bar{\mathbf{x}} > 0$. Ainsi le vecteur $\frac{1}{\mathbf{g}^T\bar{\mathbf{x}}}\bar{\mathbf{x}}$ est une solution de (3.2).

Remarquer que lorsque tous les coefficients de la contrainte de normalisation sont positifs, c'est-à-dire $\mathbf{g} \in \text{int}\mathbf{E}_+^n$, l'hypothèse (3.4) est triviale. Il y a plusieurs transformations permettant d'enchâsser le système d'équations et d'inéquations (3.1) dans un espace de projection ayant la forme (3.2), en voici quelques unes:

Une manière immédiate ou naturelle de le faire, utilisée par de Ghellinck et Vial (1986), est de constituer le système

$$\{\mathbf{M}\mathbf{x} - \mathbf{b}x_{n+1} = \mathbf{0},\ x_{n+1} = 1\ \ \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0\}$$

où x_{n+1} est une variable artificielle, dont la valeur est fixée à un de façon à rendre le système homogène. La matrice $\mathbf{A}$ a donc m lignes et $n+1$ colonnes et le vecteur $\mathbf{g} = (0, 0, \ldots, 0, 1)^T$. Toutefois cette transformation est peu souhaitable car elle ne satisfait pas nécessairement l'hypothèse (3.4). Une méthode préférable, étudiée par Vial (1989), consiste d'abord à adjoindre au système (3.1) une borne Q sur la somme des variables. Cette borne doit être suffisamment grande de manière à ne pas exclure des solutions à (3.1). A cet effet, les résultats énoncés à la section précédente peuvent être utiles. La contrainte supplémentaire est $\mathbf{1}^T\mathbf{x} + x_{n+1} = Q$. Le système transformé s'écrit comme suit:

$$\{\mathbf{M}\mathbf{x} + \mathbf{b}x_{n+2} = \mathbf{0},\ \mathbf{1}^T\mathbf{x} - Qx_{n+1} = 0,\ x_{n+2} = 1,\ \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,\ x_{n+2} \geq 0\}$$

où x_{n+1} est la variable d'écart associée à la contrainte de borne et x_{n+2} est une variable artificielle. La matrice $\mathbf{A}$ a donc $m+1$ lignes et $n+2$ colonnes:

$$\mathbf{A} = \begin{bmatrix} \mathbf{M} & \mathbf{0} & -\mathbf{b} \\ \mathbf{1}^T & -Q & 0 \end{bmatrix}$$

tandis que $\mathbf{g} = (0, 0, \ldots, 0, 1)^T$. Ce nouveau système vérifie l'hypothèse (3.4).

Karmarkar (1984) utilise aussi une borne sur la somme des variables. Toutefois, il suggère une transformation plus compliquée qui ne préserve pas la structure de la matrice **M**. Tout d'abord un changement de variables qui consiste à diviser le vecteur des inconnues par une borne sur la somme des variables est effectué:

$$z_j = \frac{x_j}{Q},\ j = 1,2,\ldots,n,\ z_{n+1} = \frac{x_{n+1}}{Q}\ .$$

Dans l'espace des variables **z**, le système (3.1) muni de la contrainte sur la somme des variables s'écrit:

$$\{Q\mathbf{M}\mathbf{z} + 0z_{n+1} = \mathbf{b},\ \mathbf{1}^T\mathbf{z} + z_{n+1} = 1\ \ \mathbf{z} \geq \mathbf{0},\ z_{n+1} \geq 0\}$$

Au lieu d'employer une variable artificielle pour rendre le système homogène, comme dans les transformations précédentes, Karmarkar emploie la somme des variables $\mathbf{1}^T\mathbf{z} + z_{n+1} = 1$ de la manière suivante:

$$\{Q\mathbf{M}\mathbf{z} - (\mathbf{1}^T\mathbf{z} + z_{n+1})\mathbf{b} = \mathbf{0},\ \mathbf{1}^T\mathbf{z} + z_{n+1} = 1,\ \mathbf{z} \geq \mathbf{0},\ z_{n+1} \geq 0\}$$

La matrice **A** a donc m lignes et $n+1$ colonnes: $\mathbf{A} = [Q\mathbf{M} - \mathbf{b}\mathbf{1}^T \quad -\mathbf{b}]$ et $\mathbf{g} = \mathbf{1}^T$. Noter que $\mathbf{b}\mathbf{1}^T$ est une matrice $m \times n$ dans laquelle chaque colonne est le vecteur **b**.

Enfin, la dernière formulation, attribuée à Todd et Burrell (voir Vial, 1989), fait aussi usage d'une contrainte de borne mais aussi d'une variable artificielle x_{n+2} afin de conserver la structure de **M**, voici les détails:

$$\{\mathbf{M}\mathbf{x} - \mathbf{b}x_{n+2} = \mathbf{0},\ \mathbf{1}^T\mathbf{x} + x_{n+1} = Q,\ x_{n+2} = 1,\ \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,\ x_{n+2} \geq 0\}$$

L'utilisation de la variable artificielle permet de réécrire les deux dernières équations de la façon suivante:

$$\{\mathbf{1}^T\mathbf{x} + x_{n+1} - Qx_{n+2} = 0,\ \mathbf{1}^T\mathbf{x} + x_{n+1} + x_{n+2} = Q + 1\}$$

La matrice **A** a donc $m+1$ lignes et $n+2$ colonnes:

$$\mathbf{A} = \begin{bmatrix} \mathbf{M} & \mathbf{0} & -\mathbf{b} \\ \mathbf{1}^T & 1 & -Q \end{bmatrix}$$

et $\mathbf{g} = \mathbf{1}^T/(Q+1)$.

Avant de passer à la description de l'algorithme de Vial, nous faisons remarquer que l'ajout d'une colonne supplémentaire à l'une ou l'autre des manipulations données ci-haut permet d'aboutir à un système ayant la forme (3.2) et pour lequel un point intérieur est facile à déterminer. Par exemple, dans la dernière formulation cela donne lieu aux équations:

$$\begin{aligned} \mathbf{M}\mathbf{x} + \mathbf{0}\mathbf{x}_{n+1} - \mathbf{b}\mathbf{x}_{n+2} - [\mathbf{M}\mathbf{1} - \mathbf{b}]\mathbf{x}_{n+3} &= \mathbf{0} \\ \mathbf{1}\mathbf{x} + \mathbf{x}_{n+1} - Q\mathbf{x}_{n+2} - [n+1-Q]\mathbf{x}_{n+3} &= 0 \\ \mathbf{1}\mathbf{x} + \mathbf{x}_{n+1} + \mathbf{x}_{n+2} + \mathbf{x}_{n+3} &= Q+3 \\ \mathbf{x} \geq \mathbf{0},\ \mathbf{x}_{n+1} \geq 0,\ \mathbf{x}_{n+2} \geq 0,\ \mathbf{x}_{n+3} &\geq 0 \end{aligned}$$

pour lequel $(\mathbf{1}^T, 1, 1, 1)$ est un point réalisable.

3.2) Une méthode générale pour résoudre le problème de décision

Dans la section précédente nous avons montré plusieurs transformations permettant d'enchâsser un polyèdre défini par des contraintes linéaires dans un espace de projection. En plus, lorsque l'hypothèse (3.4) est vérifiée, c'est le cas pour toutes les transformations à l'exception de la première, les systèmes (3.2) et (3.3) sont équivalents. Le résultat principal que nous énonçons ici est que la détermination d'un point satisfaisant (3.3) s'effectue en minimisant, à l'aide de la méthode de Newton (un rappel de cette méthode se trouve dans Bazaraa et Jarvis, 1979), une certaine fonction potentielle. Cette méthode a été initialement proposée par de Ghellinck et Vial (1987). Comme précédemment, la dimension de **A** est notée $p \times q$ pour refléter que le système original (3.1) a été mis sous la forme (3.2) en utilisant au moins une variable supplémentaire et parfois un contrainte additionnelle. Voici la fonction potentielle:

$$\psi(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x}\|}{\left(\prod_{j=1}^{q} x_j\right)^{\frac{1}{q}}} \tag{3.5}$$

Cette fonction est définie à l'intérieur de l'octant positif de E^q et elle est toujours plus grande ou égale à zéro.

> Le théorème fondamental de cette approche stipule que le système (3.3) possède une solution si et seulement si
>
> $$\inf \{\psi(\mathbf{x}) : \mathbf{x} \in \text{int}E^q_+\} = 0 .$$

Nous ne répétons pas ici la preuve qui se trouve dans Vial (1989). Cette preuve fait appel à des concepts élémentaires de l'analyse mathématique mais elle est longue. Toutefois, nous complétons l'article de Vial par une discussion des conséquences de ce théorème.

En premier lieu noter que c'est l'*infémum* et non le minimum qui est recherché. Nous rappelons que la valeur minimum, notée z^* d'une fonction bornée $h: \mathbf{V} \rightarrow \mathbf{E}^1$ où **V** est un ouvert de l'espace euclidien $\mathbf{E}^n$, fait référence à un point intérieur v^* de **V** telle que $h(\mathbf{v}^*) \leq h(\mathbf{v})\ \ \forall \mathbf{v} \in \mathbf{V}$. D'autre part, l'infémum de la fonction h, noté $\inf(h)$, est la plus grande valeur telle que $\inf(h) \leq h(\mathbf{v})\ \ \forall \mathbf{v} \in \mathbf{V}$. Remarquer que cela n'exclut pas l'existence d'un minimum.

Le problème que nous traitons en réalité est la détermination d'un point satisfaisant (3.1). Or, sous les hypothèses habituelles faites pour l'étude des programmes linéaires, les solutions recherchées doivent avoir la forme $\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b}$, $\mathbf{x_B} \geq \mathbf{0}$, $\mathbf{x_B} \neq \mathbf{0}$ et $\mathbf{x_N} = \mathbf{0}$ où [B N] est une partition de la matrice des contraintes et **B** est une matrice carrée non singulière. Les solutions de **A** ne sont donc pas des points intérieurs et pour cette raison c'est bien l'infémum de ψ que nous devons chercher et non son minimum.

Du point de vue de la mise en oeuvre informatique cela soulève la question du critère d'arrêt. Il est clair que la minimisation de ψ est stoppée lorsqu'un point intérieur $\bar{\mathbf{x}}$ avec $\mathbf{A}\bar{\mathbf{x}} = \mathbf{0}$ est trouvé. Cependant dans le cas où (3.3) n'admet pas de point intérieur comme solution, un critère d'arrêt ayant la forme $\psi(\bar{\mathbf{x}}) < \varepsilon$ ou quelque chose de semblable, doit aussi être employé. Les résultats cités à la deuxième section peuvent être appliqués ici aussi. Comme précédemment, la suite de points converge vers la solution optimale mais elle ne l'atteint pas.

Afin de faciliter l'analyse mathématique de ψ, nous introduisons la transformation logarithmique (qui est monotone croissante):

$$\phi(\mathbf{x}) \;=\; \ln(\psi(\mathbf{x}))\ .$$

Le gradient et le hessien de ϕ s'obtiennent par une application directe du théorème de la dérivation en chaîne. Noter l'identité suivante: $\ln\|\mathbf{Ax}\| = \frac{1}{2}\ln(\mathbf{x}^T\mathbf{A}^T\mathbf{Ax})$.

$$\nabla\phi(\mathbf{x}) \;=\; \frac{\mathbf{A}^T\mathbf{Ax}}{\|\mathbf{Ax}\|^2} - \frac{1}{q}\mathbf{X}^{-1}\mathbf{1} \tag{3.6}$$

$$\nabla^2\phi(\mathbf{x}) \;=\; \frac{\mathbf{A}^T\mathbf{A}}{\|\mathbf{Ax}\|^2} - 2\frac{\mathbf{A}^T\mathbf{Axx}^T\mathbf{A}^T\mathbf{A}}{\|\mathbf{Ax}\|^4} + \frac{1}{q}\mathbf{X}^{-2} \tag{3.7}$$

La matrice **X** est une matrice diagonale formée des composantes du vecteur des inconnues **x**.

Bien que ψ et ϕ soient deux fois continûment différentiables sur le domaine $\{\mathbf{x} \in \text{int}\mathbf{E}^q_+,\ \mathbf{Ax} \neq \mathbf{0}\}$ la minimisation directe de ψ avec la méthode de Newton pose deux problèmes. Le premier est la génération d'un point à l'extérieur de l'octant positif et le second est que le hessien n'est pas défini positif, ni même non singulier, en tout point du domaine (considérant le terme intermédiaire de la formule (3.7)). La première difficulté peut être traitée simplement avec un mécanisme forçant la non négativité des variables tandis que la seconde difficulté est surmontée en restreignant les directions de déplacement à l'espace nul de la matrice **A** . En effet, remarquer que le hessien de ϕ, et bien sûr celui de ψ, sont définis positifs dans l'espace nul de la matrice **A** pour tout point $\bar{\mathbf{x}}$ du domaine réalisable car

$$\mathbf{d}^T\nabla^2\phi(\bar{\mathbf{x}})\mathbf{d} \;=\; \frac{1}{n}\mathbf{X}^{-2}\mathbf{d} \qquad \forall \mathbf{d} \in \mathbf{E}^q,\ \mathbf{Ad} = \mathbf{0}\ .$$

Le théorème suivant certifie la validité des corrections envisagées.

Théorème 3.1

Etant donné un point $\bar{\mathbf{x}} \in \text{int}\mathbf{E}^q_+$, le programme mathématique

$$\text{minimiser } \{\psi(\bar{\mathbf{x}}+\mathbf{d}) : \mathbf{Ad}=\mathbf{0},\ \bar{\mathbf{x}}+\mathbf{d} \in \text{int}\mathbf{E}^q_+\}$$

dans lequel le vecteur des inconnues est $\mathbf{d}$, est strictement convexe sur le domaine de définition de ψ. De plus

$$\inf\{\psi(\bar{\mathbf{x}}+\mathbf{d}) : \mathbf{Ad}=\mathbf{0},\ \bar{\mathbf{x}}+\mathbf{d} \in \text{int}\mathbf{E}^q_+\} = 0$$

si et seulement si

$$\inf\{\psi(\mathbf{d}) : \mathbf{d} \in \text{int}\mathbf{E}^q_+\} = 0$$

Ainsi à un point $\bar{\mathbf{x}}$ donné, un pas de Newton restreint à l'espace nul de la matrice $\mathbf{A}$ appliqué à la fonction ψ est obtenu en solutionnant le programme mathématique suivant:

$$\text{minimiser } \left\{\frac{1}{2}\mathbf{d}^T\nabla^2\phi(\bar{\mathbf{x}})\mathbf{d} + \nabla\phi(\bar{\mathbf{x}})^T\mathbf{d},\ \mathbf{Ad}=\mathbf{0}\right\}$$

Afin de simplifier la notation employée pour écrire les conditions d'optimalité de ce programme, le changement de variable $\mathbf{q} = \bar{\mathbf{X}}\mathbf{d}$ est introduit:

$$\text{minimiser}\left\{\frac{1}{2}\|\mathbf{q}\|^2 - \mathbf{q}^T\mathbf{1},\ \mathbf{A}\bar{\mathbf{X}}\mathbf{q}=\mathbf{0}\right\} \tag{3.8}$$

Les conditions d'optimalité sont les équations normales:

$$\mathbf{q}-\mathbf{1}+\bar{\mathbf{X}}\mathbf{A}^T\mathbf{w} = \mathbf{0} \tag{3.9}$$

$$\mathbf{A}\bar{\mathbf{X}}\mathbf{q} = \mathbf{0} \tag{3.10}$$

où $\mathbf{w} \in \mathbf{E}^p$ est le vecteur des multiplicateurs de Lagrange. Les deux lemmes suivants caractérisent les solutions du programmes mathématique (3.8).

Lemme 3.2

Si $\mathbf{q}$ est une solution de l'équation (3.9) et $\mathbf{q} < \mathbf{1}$ alors (3.3) n'a pas de solution.

Lemme 3.3

Si $\mathbf{q}$ est une solution de l'équation (3.10) et $\mathbf{q} \geq \mathbf{0}, \mathbf{q} \neq \mathbf{0}$ alors $\bar{\mathbf{X}}\mathbf{q}$ est une solution de (3.3).

Les variables $\mathbf{q}$ sont uniquement définies par ces conditions d'optimalité:

$$\mathbf{w} = (\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\bar{\mathbf{X}}\mathbf{1}$$

$$\mathbf{q} = \mathbf{1} - \bar{\mathbf{X}}\mathbf{A}^T(\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\bar{\mathbf{X}}\mathbf{1}$$

La matrice $\mathbf{P}_{A\overline{X}} = \mathbf{I} - \overline{\mathbf{X}}\mathbf{A}^T(\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}$ projette tout vecteur dans l'espace nul de la matrice $\mathbf{A}\overline{\mathbf{X}}$. La direction de recherche est donc $\mathbf{d} = \overline{\mathbf{X}}\mathbf{q} = \overline{\mathbf{X}}\mathbf{P}_{A\overline{X}}\mathbf{1}$, c'est-à-dire la projection du vecteur $\mathbf{1}$ dans l'espace nul de la matrice $\mathbf{A}\overline{\mathbf{X}}$. Nous sommes maintenant en mesure de présenter l'algorithme de Vial.

Algorithme 2 (algorithme général pour résoudre le problème de décision 2.1)

L'initialisation consiste à déterminer un paramètre entier q qui sert à contrôler la précision numérique (voir section 2), et à mettre le compteur d'itérations k à zéro. Le point de départ est $\mathbf{x}^0 = \mathbf{1}$.

Une itération est constituée des étapes suivantes:

- Former la matrice diagonale $\overline{\mathbf{X}}$ avec les composantes du vecteur $\mathbf{x}^k$ et projeter le vecteur $\mathbf{1}$ sur l'espace nul de la matrice $\mathbf{A}\overline{\mathbf{X}}$, poser $\mathbf{q} = \mathbf{P}_{A\overline{X}}\mathbf{1}$.
- Vérifier la condition du lemme (3.2). Terminer si $\mathbf{q}$ est plus petit que $\mathbf{1}$ car le système (3.3) n'admet pas de solution. Sinon, terminer si la condition du lemme (3.3) est satisfaite car $\mathbf{q}$ est un point intérieur de l'octant positif qui solutionne (3.3).
- Prendre un pas $\overline{\alpha}$ dans la direction de recherche $\mathbf{d} = \overline{\mathbf{X}}\mathbf{q}$ qui minimise $\mathbf{x}^k + \alpha\mathbf{d}$ de façon à rester dans l'intérieur de l'octant positif. Poser $\mathbf{x}^{k+1} = \mathbf{x}^k + \overline{\alpha}\mathbf{d}$ et mettre à jour k.

La procédure termine quand $\| \mathbf{A}\mathbf{x}^k \| < 2^{-q}$.

L'algorithme 2 est une procédure itérative qui engendre une suite de points intérieurs à l'octant positif $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^k$ qui converge vers la solution optimale. Toutefois, ces points ne sont pas réalisables pour (3.3) sauf quand la condition du lemme (3.3) est satisfaite et l'algorithme termine. La preuve de convergence est aussi basée sur la réduction, à chaque itération, de la fonction ϕ par au moins une quantité constante. La preuve emploie les mêmes arguments que ceux présentés à la fin de la première section et les détails se trouvent dans de Ghellinck et Vial (1986). Il est intéressant de remarquer que la procédure ne fait pas usage de transformations projectives car elle opère directement dans l'espace de projection. Pour cette raison, Vial qualifie l'algorithme 2 d'algorithme de projection. Dans ce qui suit nous indiquons l'équivalence avec l'algorithme de Karmarkar.

3.3) Equivalences avec la méthode de Karmarkar

Rappelons que le programme canonique de Karmarkar consiste à minimiser une fonction linéaire dont la valeur optimale est zéro sur le domaine réalisable formé par l'intersection d'un sous-espace linéaire et du simplexe unité. Plusieurs auteurs (Gonzaga, 1989; Vial, 1989; Todd, 1988) soulignent que l'algorithme de Karmarkar n'est pas intuitif et que le rôle joué par le simplexe unité, où l'action se déroule, n'est pas clair. Dans ce qui suit, nous faisons ressortir que la méthode de Karmarkar est une spécialisation de l'algorithme 2. Cela permet de conclure que les opérations

de projection et la présence du simplexe unité dans l'algorithme de Karmarkar sont inutiles. Dans ce qui suit, la matrice des contraintes du programme canonique (1.1) est noté $\mathbf{A}$ tandis que la matrice de coefficients de (3.3) est $\tilde{\mathbf{A}}$.

L'algorithme de Karmarkar produit une suite de points qui sont strictement positifs et qui appartiennent à l'espace nul de la matrice $\mathbf{A}$ des contraintes du programme canonique (1.1). Puisque l'algorithme 1 assume que la valeur optimale du programme canonique est zéro et puisque la présence du simplexe unité assure que l'hypothèse (3.4) est satisfaite, cela revient à déterminer un rayon positif non trivial $\mathbf{x} \in \mathbf{E}^q$, $\mathbf{x} \geq \mathbf{0}$, $\mathbf{x} \neq \mathbf{0}$ satisfaisant le système (3.3) ayant $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{c}^T\mathbf{x} \\ \mathbf{A} \end{bmatrix}$ comme matrice des coefficients des équations homogènes. Noter que le programme canonique de Karmarkar est en réalité un programme entièrement homogène.

Une spécialisation de l'algorithme 2 pour résoudre ce problème consiste à restreindre le vecteur des inconnues $\mathbf{x}$ au cône $C = \{z \in \mathbf{E}^q : \mathbf{Aq} = \mathbf{0},\ \mathbf{q} > \mathbf{0}\}$. Cette spécialisation de l'algorithme 2 produit une séquence de points qui appartiennent au domaine réalisable du programme canonique de Karmarkar. Pour ce faire, la fonction potentielle ψ restreinte aux rayons de ce cône est:

$$\tilde{\psi}(\mathbf{x}) = \frac{\left\| \begin{bmatrix} \mathbf{c}^T \\ \mathbf{A} \end{bmatrix} \mathbf{x} \right\|}{\left(\prod_{j=1}^{q} x_j \right)^{1/q}} = \frac{\mathbf{c}^T\mathbf{x}}{\left(\prod_{j=1}^{q} x_j \right)^{1/q}} \tag{3.11}$$

En outre, le logarithme de $\tilde{\psi}(\mathbf{x})$ est équivalent à la fonction potentielle employée par Karmarkar.

En second lieu, la direction de recherche de l'algorithme 2 est la projection du vecteur $\mathbf{1}$ sur l'espace nul de la matrice $\tilde{\mathbf{A}}$. Etant donné un point intérieur $\bar{\mathbf{x}}$, voici la forme explicite de la direction de recherche, notée $\mathbf{d}_V$ (V pour Vial), quand le vecteur des inconnues est restreint au cône C:

$$\mathbf{d}_V = \bar{\mathbf{x}} - \frac{\mathbf{c}^T\bar{\mathbf{x}}}{\|\mathbf{P}_{A\bar{X}}\bar{\mathbf{X}}\mathbf{c}\|^2}\bar{\mathbf{X}}\mathbf{P}_{A\bar{X}}\bar{\mathbf{X}}\mathbf{c}\ . \tag{3.12}$$

L'obtention de cette formule requiert deux lemmes techniques qui sont démontrés par Vial et dont la présentation ici ne ferait qu'allonger inutilement la discussion.

Maintenant, revisons le calcul de la direction de recherche de l'algorithme 1. Un point intérieur $\bar{\mathbf{x}}$ satisfaisant les contraintes de programme (1.1) est donné. La direction de recherche, notée $\mathbf{d}_K$ (K pour Karmarkar), dans l'espace transformé, est la projection du vecteur $\bar{\mathbf{X}}\mathbf{c}$ dans l'espace nul de la matrice $\begin{bmatrix} \mathbf{A}\bar{\mathbf{X}} \\ \mathbf{1}^T \end{bmatrix}$. D'après les formules (1.10)-(1.12) la direction de recherche dans l'espace transformé est $\mathbf{d}_K = -\bar{\mathbf{X}}(\mathbf{c} - \mathbf{A}^T\mathbf{u}) + \mathbf{c}^T\bar{\mathbf{x}}\mathbf{1}/n$ où $(\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)\mathbf{u} = \mathbf{A}\bar{\mathbf{X}}^2\mathbf{c}$. Afin de faciliter notre analyse nous remplaçons $\mathbf{u}$ par sa valeur (la matrice $\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T$ est définie positive et de rang plein) et nous écrivons

$$\mathbf{d}_K = -\bar{\mathbf{X}}\mathbf{c} + \bar{\mathbf{X}}\mathbf{A}^T(\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\bar{\mathbf{X}}^2\mathbf{c} + \mathbf{c}^T\bar{\mathbf{x}}\mathbf{1}/n \tag{3.13}$$

Dans l'espace original, la direction de recherche s'obtient en pré-multipliant le vecteur $\mathbf{d}_K$ par la matrice $\overline{\mathbf{X}}$:

$$\overline{\mathbf{X}}\mathbf{d}_K = -\overline{\mathbf{X}}^2\mathbf{c} + \overline{\mathbf{X}}^2\mathbf{A}^T(\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}^2\mathbf{c} + \overline{\mathbf{X}}\mathbf{c}^T\overline{\mathbf{x}}\mathbf{1}/n \tag{3.14}$$

Quelques manipulations algébriques supplémentaires donnent lieu à la formule suivante:

$$\overline{\mathbf{X}}\mathbf{d}_K = -\overline{\mathbf{X}}\mathbf{P}_{A\overline{X}}\overline{\mathbf{X}}\mathbf{c} + \frac{\mathbf{1}^T\mathbf{P}_{A\overline{X}}\overline{\mathbf{X}}\mathbf{c}}{n}\overline{\mathbf{x}}$$

Une caractéristique d'un espace de projection est qu'il est constitué de rayons. Deux vecteurs appartenant au même rayon sont donc équivalents à cause de la contrainte de normalisation. Dans le cas des algorithmes qui opèrent sur un espace de projection, la comparaison de deux directions de recherche consiste à vérifier que les cônes correspondant soient identiques. Le cône convexe de dimension deux qui représente tous les rayons qui peuvent être générés à paritr d'un point $\overline{\mathbf{x}}$ et d'une direction de recherche $\mathbf{d}$ est

$$C(\overline{\mathbf{x}}, \mathbf{d}) = \{\mathbf{z} \in \mathbf{E}^q,\ \mathbf{z} = \alpha\overline{\mathbf{x}} + \beta\mathbf{d},\ \alpha \geq 0,\ \beta \geq 0,\ \alpha\overline{\mathbf{x}} + \beta\mathbf{d} \geq \mathbf{0}\}$$

Vial (1989) a montré que $C(\overline{\mathbf{x}}, \mathbf{d}^V) = C(\overline{\mathbf{x}}, \overline{\mathbf{X}}\mathbf{d}^K)$.

3.4) Autres variantes et extensions

Supposer que la valeur optimale d'un programme linéaire soit connue est très restrictif. Pour cette raison, nous avons jugé nécessaire d'analyser l'algorithme de Karmarkar en adoptant le point de vue des problèmes de décision. Nous terminons cette troisième section en pointant des extensions et des variantes à l'algorithme de Karmarkar qui ont été mises au point dans le but de contourner cette restriction.

A l'aide d'une des transformations présentée au début de cette section, un programme linéaire général admet la forme suivante:

$$\text{minimiser } \{\tilde{\mathbf{c}}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0},\ \mathbf{g}^T\mathbf{x} = \mathbf{1},\ \mathbf{x} \geq \mathbf{0}\}$$

Cependant, nous ne faisons aucune hypothèse sur la valeur optimale z^* sauf que $\underline{z}$ est une borne inférieure connue de z^*. La minimisation de ce programme repose sur la détermination d'un point qui satisfait le système

$$\{\mathbf{x} \in \mathbf{E}^q : (\tilde{\mathbf{c}} - \underline{z}\mathbf{g})^T\mathbf{x},\ \mathbf{A}\mathbf{x} = \mathbf{0},\ \mathbf{x} \geq \mathbf{0},\ \mathbf{x} \neq \mathbf{0}\}$$

qui dépend du paramètre $\underline{z}$. L'algorithme 2 peut être modifié directement pour permettre la mise à jour périodique de $\underline{z}$. Cette modification repose sur l'observation suivante: lorsque l'algorithme termine ou bien $\underline{z} = z^*$, ou bien le système n'admet pas de solution. Pour ce dernier cas, Vial (1989) montre les formules nécessaires pour la mise à jour du paramètre. L'avantage est que l'algorithme peut poursuivre avec le point courant sans devoir recommencer à zéro.

D'autres méthodes ont aussi été proposées par Anstreicher (1986) qui utilise des bornes obtenues à l'aide des propriétés géométriques tandis que Todd et Burrell (1986) utilisent la théorie de la dualité. Vial (1989) a d'ailleurs montré que sa méthode peut être rendue équivalente à celle de Todd et Burrell (1986). Cependant, le traitement direct des programmes linéaires donnés sous la forme standard s'effectue en pratique avec les algorithmes usant de transformations affines et les méthodes de trajectoire.

4) METHODES AFFINES

A chaque itération de l'algorithme de Karmarkar, un point intérieur appartenant à l'intersection d'un cône et du simplexe unité est employée dans un changement de variables qui possède la propriété de projeter le simplexe unité sur lui-même et le point intérieur au centre de ce simplexe. Dans la section précédente nous avons fait ressortir que la présence du simplexe unité et l'opération de géométrie projective reviennent à employer un algorithme qui opère directement dans l'espace de projection. Nous avons établi que le programme canonique de Karmarkar possède en réalité une structure entièrement homogène et le point de vue adopté a été celui des problèmes de décision. Des transformations et des extensions pour traiter les programmes linéaires donnés sous la forme standard ont aussi été mentionnées.

Nous poursuivons notre revue des algorithmes de points intérieurs avec les travaux de plusieurs chercheurs, en particulier Barnes (1986), Vanderbei et al. (1986) et Adler et al. (1989), qui ont mis au point des méthodes usant de transformations affines au lieu des transformations projectives. L'avantage des méthodes affines repose sur le traitement direct des programmes linéaires données sous la forme standard. Il y a aussi un lien direct avec les algorithmes de type gradient projeté. D'un autre coté, l'absence d'une mesure permettant de déduire la convergence en un nombre d'itérations qui dépend d'un polynôme a été observé. Les mises en oeuvre informatique sont cependant très efficaces (voir conclusion).

Goldfarb et Todd (1989) mentionnent que l'étude des méthodes affines précède d'une décennie l'algorithme de Karmarkar et est due au mathématicien russe Dikin [1]. Ces algorithmes consistent à appliquer la méthode du gradient projeté dans un espace transformé par une application affine. Cependant, afin d'éviter d'aboutir aux frontières du domaine réalisable définies par les contraintes d'inégalité, une fraction du pas maximal permis est employée. Barnes (1986) et Vanderbei et al. (1986) appliquent cette technique aux programmes linéaires donnés sous la forme standard, c'est la version primale, tandis que Adler et al. (1989) ont considéré la version duale.

En premier lieu, nous donnons les détails de l'algorithme affine primal. Soit le programme linéaire donné sous la forme standard

$$\text{minimiser } \{\mathbf{c}^T\mathbf{x}:\ \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq\mathbf{0}\} \tag{4.1}$$

pour lequel un point intérieur $\overline{\mathbf{x}}$ est connu. Ecrivons tout de suite le dual de (4.1).

$$\text{maximiser } \{\mathbf{b}^T\mathbf{u}:\ \mathbf{A}^T\mathbf{u}+\mathbf{v}=\mathbf{c},\ \mathbf{v}\geq\mathbf{0}\} \tag{4.2}$$

La transformation affine $\mathbf{y}=\overline{\mathbf{X}}^{-1}\mathbf{x}$ qui admet la transformation inverse unique $\mathbf{x}=\overline{\mathbf{X}}\mathbf{y}$ donne lieu au programme linéaire équivalent dans l'espace des $\mathbf{y}$:

$$\text{minimiser } \{\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y}:\ \mathbf{A}\overline{\mathbf{X}}\mathbf{y}=\mathbf{b},\ \mathbf{y}\geq\mathbf{0}\} \tag{4.3}$$

---

[1] Dikin I., 1967, *Iterative Solution of Problems of Linear and Quadratic Programming*, **Soviet Mathematics Doklady**, 8, 674-675.

Noter que dans l'espace transformé le point intérieur $\bar{\mathbf{x}}$ correspond au vecteur $\mathbf{1}$. Dans cet espace, la direction réalisable de la plus forte pente est la projection du vecteur $\bar{\mathbf{X}}\mathbf{c}$, c'est le gradient de la fonction objectif de (4.3), sur l'espace nul de la matrice $\mathbf{A}\bar{\mathbf{X}}$. Pour déterminer cette direction de recherche un vecteur $\mathbf{u}$ qui solutionne le système d'équations linéaires $(\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)\mathbf{u} = \mathbf{A}\bar{\mathbf{X}}^2\mathbf{c}$ est calculé (voir section 1). Ainsi, la direction de recherche dans l'espace transformé est $\mathbf{d} = -\bar{\mathbf{X}}(\mathbf{c} - \mathbf{A}^T\mathbf{u})$ et dans l'espace original $\mathbf{d} = -\bar{\mathbf{X}}^2(\mathbf{c} - \mathbf{A}^T\mathbf{u})$.

Par construction, $\mathbf{d}$ est une direction de descente réalisable et le nouveau point est $\bar{\mathbf{x}} + \alpha\mathbf{d}$, $0 < \alpha < \alpha_{max}$. Le pas maximal assure que la nouvelle solution demeure dans l'octant positif, c'est-à-dire $\alpha_{max} = \max\{\alpha > 0:\ \bar{\mathbf{x}} + \alpha\mathbf{d} \geq \mathbf{0}\}$:

$$\alpha_{max} = \min_j \left\{ -\frac{\bar{x}_j}{d_j} :\ d_j < 0 \right\} \tag{4.4}$$

Remarquer que la fonction à minimiser est linéaire, c'est donc le pas maximal qui minimise $\mathbf{c}^T(\bar{\mathbf{x}} + \alpha\mathbf{d})$. Dans l'esprit des méthodes de points intérieurs, contrairement à la méthode du gradient projeté, un pas inférieur au pas maximal est choisi. D'une façon formelle l'algorithme affine primal est le suivant:

Algorithme 3 (algorithme affine primal)

L'initialisation consiste à déterminer un point intérieur $\mathbf{x}^0$ du domaine réalisable de (4.1) et à choisir une proportion γ, $0 < \gamma < 1$ du pas maximal. Le compteur d'itérations k est aussi mis à zéro.

Une itération est constituée des étapes suivantes:

- Résoudre le système d'équations $(\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^T)\mathbf{u} = \mathbf{A}\bar{\mathbf{X}}^2\mathbf{c}$ où le vecteur des inconnues est $\mathbf{u}$ et où $\bar{\mathbf{X}}$ est la matrice diagonale formée des composantes de la solution courante $\mathbf{x}^k$.
- Dans l'espace original, la direction de recherche est $\mathbf{d} = \bar{\mathbf{X}}^2(\mathbf{c} - \mathbf{A}^T\mathbf{u})$. Le programme (4.1) n'est pas borné inférieurement quand $\mathbf{d} > \mathbf{0}$; il faut alors terminer. Sinon, déterminer le pas maximal α_{max} selon la formule (4.4) et poser $\alpha = \gamma \cdot \alpha_{max}$.
- La nouvelle solution est $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \cdot \mathbf{d}$.

La procédure termine quand la diminution de la valeur objectif entre deux itérations est très faible en pourcentage.

Cet algorithme appelle quelques commentaires. Tout d'abord, la suite $\{\mathbf{c}^T\mathbf{x}^k\}$ est monotone décroissante:

$$\mathbf{c}^T\mathbf{x}^k - \mathbf{c}^T\mathbf{x}^{k+1} = \mathbf{c}^T\mathbf{x}^k - \mathbf{c}^T(\mathbf{x}^k + \alpha\mathbf{d}) = -\alpha\mathbf{c}^T\mathbf{d} = \alpha\|\mathbf{d}\|^2 > 0$$

L'algorithme peut donc être stoppé quand la diminution de la valeur objectif entre deux itérations est très faible. Aussi, dans le cas où la direction de recherche est plus grande ou égale à zéro, $\mathbf{u}$ est une solution réalisable pour le programme linéaire dual (4.2) et $\mathbf{b}^T\mathbf{u}$ constitue une borne inférieure sur la valeur de la solution optimale. Cela permet de construire un autre critère d'arrêt.

Les frontières du domaine réalisable de (4.1) proviennent des inégalités $\mathbf{x} \geq \mathbf{0}$. La transformation affine a pour but d'éloigner $\bar{\mathbf{x}}$ des frontières. Dans le nouveau système de coordonnées $\bar{\mathbf{x}}$ est à une unité de chacune des frontières. Le rôle du paramètre γ est de maintenir la suite de solutions générées dans l'intérieur de l'octant positif. Une manière élégante de déterminer la longueur du pas afin de ne pas aboutir à une frontière a été proposée par Barnes (1986). L'ellipsoïde centrée à $\bar{\mathbf{x}}$

$$E(\bar{\mathbf{x}},\gamma) = \sum_{j=1}^{n}\left[\frac{x_j-\bar{x}_j}{\bar{x}_j}\right]^2 \leq \gamma^2 \qquad 0<\gamma<1$$

se situe entièrement dans l'octant positif. Le point minimum de la fonction linéaire $\mathbf{c}^T\mathbf{x}$ dans le domaine réalisable défini par l'intersection de l'ellipsoïde $E(\bar{\mathbf{x}},\gamma)$ avec les contraintes affines $\mathbf{Ax} = \mathbf{b}$ est

$$\mathbf{x}^*_{\bullet} = \bar{\mathbf{x}} - \gamma \frac{\bar{\mathbf{X}}^2(\mathbf{c}-\mathbf{A}^T\mathbf{u})}{\|\bar{\mathbf{X}}(\mathbf{c}-\mathbf{A}^T\mathbf{u})\|} .$$

Le dénominateur du terme de droite est la norme de la direction de recherche dans l'espace transformé tandis que le numérateur est la direction de recherche dans l'espace original. Cela suggère l'emploi du pas $\alpha = \gamma/\|\bar{\mathbf{X}}(\mathbf{c}-\mathbf{A}^T\mathbf{u})\|$. Dans ce cas le paramètre γ mesure la proximité permise de la nouvelle solution avec les frontières de l'octant positif. Nous faisons remarquer l'analogie avec l'algorithme de Karmarkar qui à chaque itération obtient une nouvelle solution en minimisant une fonction linéaire ayant pour domaine réalisable l'intersection d'une boule (au lieu d'une ellipsoïde) avec un sous-espace de projection (au lieu d'un sous-espace affine).

Une seconde manière de mettre au point une méthode de transformation affine est d'appliquer l'algorithme 3 au programme linéaire dual (4.2). Cela a été étudié par Adler et al. [1] (voir Goldfarb et Todd, 1989). Des mises en oeuvre informatiques ont été publiées par Adler et al. (1989) et Marsten et al. (1989).

Ici, les contraintes d'inégalités (quand (4.2) est écrit sans les variables d'écart, c'est-à-dire $\mathbf{A}^T\mathbf{u} \leq \mathbf{c}$) et le fait que $\mathbf{u}$ ne soit pas restreint à l'octant positif présentent l'avantage qu'un calcul inexact de la direction de recherche ne met pas en cause la faisabilité de la nouvelle solution. L'algorithme est le suivant:

---

[1] Adler I., Karmarkar N., Resende M., Veiga G., 1986, *An Implementation of Karmarkar's Algorithm for Linear Programming*, Working Paper, Operations Research Center, University of California, Berkeley, CA.

Algorithme 4 (algorithme affine dual)

L'initialisation consiste à déterminer un point réalisable $\mathbf{u}^0$ de (4.2) et à choisir une proportion γ, $0 < \gamma < 1$. Le compteur d'itérations k est aussi mis à zéro.

Une itération est constituée des étapes suivantes:

- Calculer la valeur des variables d'écart: $\mathbf{v}^k = \mathbf{c} - \mathbf{A}^T\mathbf{u}^k$. Résoudre le système d'équations $(\mathbf{A}\overline{\mathbf{V}}^{-2}\mathbf{A}^T)\mathbf{d} = \mathbf{b}$ où le vecteur des inconnues est $\mathbf{d}$ et où $\overline{\mathbf{V}}$ est la matrice diagonale formée des composantes des variables d'écart.
- Le programme (4.2) n'est pas borné supérieurement quand $-\mathbf{A}\overline{\mathbf{d}} \geq \mathbf{0}$; il faut alors terminer. Sinon, poser $\alpha = -\gamma/\min_j\{(\overline{\mathbf{V}^{-1}\mathbf{d}})_j\}$.
- La nouvelle solution est $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha \cdot \mathbf{d}$.

La procédure termine quand la diminution de la valeur objectif entre deux itérations est très faible en pourcentage.

Des preuves de convergence des algorithmes 3 et 4 se trouvent dans Barnes (1986) et Vanderbei et al. (1986). Toutefois, contrairement à l'algorithme de Karmarkar, la réduction d'une fonction mesurant le progrès vers la solution optimale par une constante n'est pas garanti. Dans la prochaine section nous mentionnons une variantes dont la complexité théorique dépend d'un polynôme. C'est une procédure à cheval sur les méthodes de transformations affines et les méthodes de trajectoire.

5) METHODES DE TRAJECTOIRE

Nous avons vu que les algorithmes de points intérieurs usant de transformations affines sont des spécialisations de la méthode du gradient projeté dans laquelle un mécanisme empêchant les solutions d'aboutir aux frontières est utilisé. Ces algorithmes convergent vers la solution optimale mais le nombre d'opérations élémentaires n'est pas borné par un polynôme. Récemment, Gonzaga (1988) et Ye (1988) ont montré qu'en calculant la direction de recherche avec le gradient d'une fonction potentielle de type logarithme (à la place du gradient d'une fonction linéaire) l'algorithme 3 a une complexité théorique qui dépend d'un polynôme. La fonction potentielle a la forme suivante:

$$\phi(\mathbf{x},\underline{z}) \;=\; q\ln(\mathbf{c}^{\mathrm{T}}\mathbf{x}-\underline{z}) \;-\; \sum_{j=1}^{n} x_j$$

où $q=n+\sqrt{n}$ et $\underline{z}$ est une borne inférieure. Cette variante de l'algorithme 3 combine donc des concepts des algorithmes avec transformations projectives (fonction potentielle) et des algorithmes avec transformation affines (traitement direct des programmes linéaires donnés sous la forme standard et méthode du gradient projeté).

Dans cette section étudions les ressemblances entre les fonction potentielle et les fonctions de barrière de Fiacco et McCormick. Les algorithmes étudiés ici appartiennent à la classe des méthodes de trajectoire. Elles allient plusieurs techniques bien connues de l'optimisation classique et elles reposent sur des propriétés analytiques de l'intérieur des polyèdres convexes. Pour cette raison nous pensons que l'analyse de ces algorithmes constitue un cadre unificateur pour l'étude des programmes linéaires.

Les fonctions de barrière de type logarithme ont été proposées pour la première fois par Frish en 1955 dans un document de travail de l'université d'Oslo. Des notes historiques intéressantes se trouvent dans le volume de Fiacco et McCormick (1968). D'une façon générale, l'utilisation d'une fonction de barrière sert à transformer un programme mathématique ayant la forme

$$\text{minimiser } \{f(\mathbf{x}):\ g_i(\mathbf{x})\geq 0,\ i=1,2,\ldots,m\} \tag{5.1}$$

en une famille de fonctions à minimiser en introduisant un mécanisme naturel qui force les contraintes de non négativité à être satisfaites. Cela s'accompli à l'aide d'une fonction auxiliaire qui dépend d'un paramètre positif μ:

$$h(\mathbf{x}\,|\,u) \;=\; f(\mathbf{x})+\mu\sum_{i=1}^{m} B(g_i(\mathbf{x})) \tag{5.2}$$

où $B(y)$ est continu pour tout $y>0$ et $\lim_{y\to 0} B(y)=\infty$. Fiacco et McCormick (1968) donnent une analyse détaillée de cette approche. Un résultat important montré dans ce livre est que sous certaines hypothèses de régularité, la solution $\mathbf{x}(\mu)$ qui minimise (5.2) converge vers la solution optimale de (5.1) quand μ décroît vers zéro. Noter que pour tout $\mu>0$, $\mathbf{x}(\mu)$ appartient au domaine réalisable de (5.1). Ainsi, le "refroidissement" du paramètre μ permet un rapprochement vers les frontières du domaine réalisable.

Dans le contexte de l'étude des programmes linéaires Meggido (1989) et Bayer et Lagarias (1987a, 1987b) ont analysé en profondeur les propriétés des fonctions de barrière de type logarithme. Soulignons qu'une analyse très détaillée requiert des connaissances mathématiques très poussées (métrique de Riemann, courbes algébriques, équations différentielles, transformation de Legendre). Pour cette raison, nous résumons les points importants et nous référons le lecteur aux articles cités ci-haut pour les détails et d'autres résultats. Mentionnons aussi le nombre élevé d'algorithmes qui appartiennent à la classes des méthode de trajectoire. Cependant, les différences sont parfois très subtiles.

Soit le couple de programmes linéaires suivant. Le premier est le programme primal et le second est son dual:

$$\text{minimiser } \{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq\mathbf{0}\} \tag{5.3}$$

$$\text{maximiser } \{\mathbf{b}^T\mathbf{u} : \mathbf{A}^T\mathbf{u}+\mathbf{v}=\mathbf{c},\ \mathbf{v}\geq\mathbf{0}\} \tag{5.4}$$

Le vecteur $\mathbf{v}$ est le vecteur des variables d'écart. L'utilisation d'une fonction de barrière de type logarithme donne lieu à deux familles de programmes mathématiques, une pour le primal et une pour le dual, qui dépendent du paramètre positif μ:

$$\text{minimiser } \left\{\mathbf{c}^T\mathbf{x}-\mu\sum_{j=1}^{n}\ln x_j : \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}>\mathbf{0}\right\} \tag{5.5}$$

$$\text{maximiser } \left\{\mathbf{b}^T\mathbf{u}-\mu\sum_{j=1}^{n}\ln v_j : \mathbf{A}^T\mathbf{u}+\mathbf{v}=\mathbf{c},\ \mathbf{v}>\mathbf{0}\right\} \tag{5.6}$$

En supposant que l'intérieur du domaine réalisable du programme primal (5.3) ne soit pas vide et qu'il admette une solution optimale finie, la fonction de barrière (5.5) a une solution optimale unique $\mathbf{x}(\mu)$ pour tout $\mu>0$ qui converge vers une solution optimale de (5.3) quand μ approche zéro. Cela implique qu'il existe un couple $(\ \mathbf{u}(\mu), \mathbf{v}(\mu)\ )$ qui est une solution optimale de (5.6) qui converge vers la solution optimale de (5.4). Les conditions nécessaires d'optimalité stipulent que le triplet $(\ \mathbf{x}(\mu), \mathbf{u}(\mu), \mathbf{v}(\mu)\ )$ satisfait les équations suivantes:

$$\mathbf{UX1} - \mu\mathbf{1} = \mathbf{0}, \tag{5.6}$$

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0}, \quad \mathbf{x}>\mathbf{0}, \tag{5.7}$$

$$\mathbf{A}^T\mathbf{u} + \mathbf{v} - \mathbf{c} = \mathbf{0}, \tag{5.8}$$

où $\mathbf{X}$ et $\mathbf{U}$ sont des matrices diagonales formées des composantes des vecteurs $\mathbf{x}$ et $\mathbf{u}$ respectivement.

Gill et al (1986) ont été les premiers à soulever les équivalences entre les méthodes de l'optimisation classique et l'algorithme de Karmarkar. Ils ont montré que résoudre (5.5) avec un choix particulier de μ à l'aide de l'algorithme de Newton (dans lequel la direction de recherche est restreinte à l'espace nul de la matrice $\mathbf{A}$ pour tenir compte des contraintes d'égalité) correspond à l'algorithme de Karmarkar.

En considérant des variations infinitésimales de μ, il est possible de comparer $\mathbf{x}(\mu)$ à une courbe algébrique qui définit une trajectoire, appelée *trajectoire centrale*, dans le domaine réalisable vers la solution optimale. L'étude des propriétés de cette trajectoire fait appel aux équations différentielles. Il y a trois avantages reliés à la mise en oeuvre d'un algorithme dont les points générés suivent la trajectoire centrale: près de la solution optimale la trajectoire est presqu'une ligne droite, la complexité théorique de ces méthodes est polynomiale et des extensions efficaces tant du point de vue pratique que théorique au cas où la fonction objectif est une fonction convexe sont aussi possibles (voir Monteiro et Adler, 1989b). Toutefois, la convergence n'est pas très rapide car suivre la trajectoire centrale nécessite des faibles variations de la valeur du paramètre μ.

En pratique, les méthodes de barrière pour solutionner des programmes mathématiques ayant la forme (5.1) emploient des grands pas afin d'accélérer la convergence. L'algorithme de Gill et al. (1986), par exemple, appartient à cette catégorie. Cependant, les méthodes de trajectoire produisent une suite de points qui sont "près" de la trajectoire centrale selon une mesure que nous précisons plus loin.

Les méthodes de trajectoire se divisent en trois groupes: suivre la trajectoire centrale dans le domaine réalisable du programme linéaire primal; suivre la trajectoire centrale dans le domaine réalisable du programme linéaire dual et suivre la trajectoire centrale dans le domaine réalisable formé par les conditions de complémentarité, ce sont les méthodes primales-duales.

Ye (1988) mentionne que les méthodes du premier groupe ont été étudiées par Goldfarb et Liu [1] et Ye [2]. Ces méthodes conviennent aussi pour la minimisation de fonctions convexes ayant pour domaine réalisable des contraintes linéaires. Une description de ce type algorithme avec une analyse détaillée de la convergence se trouve dans Goldfarb et Todd (1989).

Renegar (1988) a été le premier a suggérer un algorithme générant une suite de points proche de la trajectoire centrale. Son algorithme appartient à la seconde catégorie. Cependant, la procédure repose sur une autre technique classique de l'optimisation appelée *méthode des centres*. Etant donné un polyèdre $P = \{\mathbf{x} \in \mathbf{E}^n: \ \mathbf{Ax} \geq \mathbf{b}\}$ et la fonction potentielle

$$h(\mathbf{x}) \ = \ \sum_{i=1}^{m} \ln(a_i x_j - b_i)$$

le *centre analytique* de P est un point intérieur $\bar{\mathbf{x}}$, c'est-à-dire $\mathbf{A}\bar{\mathbf{x}} > \mathbf{b}$, qui maximise $h(\mathbf{x})$. L'algorithme de Renegar détermine le maximum d'une fonction linéaire ayant P comme domaine réalisable et il repose sur la détermination successive des centres analytiques de la famille de polyèdres $\{\mathbf{x} \in \mathbf{E}^n: \ \mathbf{c}^T\mathbf{x} \geq \mu, \ \mathbf{Ax} \geq \mathbf{b}\}$ qui dépend du paramètre μ. La méthode des centres est due à Huard [3]

---

[1] Goldfarb D., Liu S., 1988, *An $O(n^3L)$ Primal Interior Point Algorithm for Convex Quadratic Programming*, Technical Report, Department of IEOR, Columbia University, New York, NY.

[2] Ye Y., 1987, *Interior Algorithms for Linear, Quadratic, and Linearly Constrained Convex Programming*, Ph.D. Thesis, Department of Engineering-Economic Systems, Stanford University, Standford, CA.

[3] Huard P., 1967, *Resolution of Mathematical Programming with Nonlinear Constraints by the Method of Centers*, dans J. Abadie, éditeur, **Nonlinear Programming**, North-Holland, Amsterdam.

(voir Fiacco et McCormick) et elle sert à résoudre des programmes convexes. Bayer et Lagarias (1987a) ont établi les équivalences entre les centres analytiques et la trajectoire $\mathbf{x}(\mu)$ induite par l'utilisation des fonctions de barrière de type logarithme. Ces équivalences sont à l'origine du nom *trajectoire centrale*. Un autre algorithme qui suit la trajectoire duale se trouve dans Gonzaga (1989).

Les algorithmes de trajectoire les plus populaires appartiennent à la troisième catégorie. Ceux-ci ont été étudiés par Kojima et al. (1989), Lustig et al. (1990a,b), Monteiro et Adler (1989a,b), Todd et Ye (1990). Des mises en oeuvre informatique sont présentées dans McShane et al. (1989) et Lustig et al. (1990c). Enfin, Marsten et al. (1990) en donne une description très intuitive.

Un algorithme de trajectoire primal-dual consiste à résoudre, pour différentes valeurs du paramètre μ qui approchent zéro, les équations (5.6)-(5.8). La suite $\{\mathbf{x}(\mu)\}$ converge vers la solution optimale du programme primal tandis que suite $\{\mathbf{u}(\mu), \mathbf{v}(\mu)\}$ converge vers la solution optimale du programme dual.

Un point satisfaisant les équations des conditions d'optimalité est calculé à l'aide de la méthode de Newton. De plus, le paramètre μ est modifié à chaque fois qu'un pas dans la direction de Newton est pris au lieu d'attendre qu'une solution soit trouvé. Ceci accélère la convergence mais le nouveau point s'éloigne de la trajectoire centrale. Dans ce cas la notion de proximité avec la trajectoire centrale est introduite. Un triplet $(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{v}})$ qui satisfait (5.7)-(5.8) et qui vérifie l'inégalité

$$\| \overline{\mathbf{XV}}\mathbf{1} - \bar{\zeta}\mathbf{1} \| \leq \alpha\bar{\zeta}$$

où $\bar{\zeta} = \mathbf{1}^T\overline{\mathbf{XV}}\mathbf{1}/n$ et $0 < \alpha < 1$ est "près" de la trajectoire centrale. Noter que l'équation (5.6) des conditions d'optimalité est non linéaire. Une solution approximative est suffisante. Le nouveau point est $(\bar{\mathbf{x}} + \mathbf{d}_x, \bar{\mathbf{u}} + \mathbf{d}_u, \bar{\mathbf{v}} + \mathbf{d}_v)$ où la direction est obtenue avec les formules suivantes:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{V} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_u \\ \mathbf{d}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mu\mathbf{1} - \overline{\mathbf{XV}}\mathbf{1} \end{bmatrix} \tag{5.9}$$

Les différents algorithmes se distinguent par la formule utilisée pour la modification de μ. Par exemple, Monteiro et Adler (1989) ont montré que le nouveau point est près de la trajectoire centrale quand $\mu = (1 - \gamma/\sqrt{n})\bar{\zeta}$ pour une constante positive γ.

CONCLUSION

Dans ce travail nous avons adopté le point de vue de plusieurs chercheurs qui classifient en deux groupes les méthodes de points intérieurs pour la résolution des programmes linéaires.

Le premier groupe comprend l'algorithme de Karmarkar, ses variantes et ses extensions. Parmi ces algorithmes nous avons fait la distinction entre ceux qui emploient des transformations projectives et ceux qui emploient des transformations affines. Karkarkar décrit le fonctionnement de ces algorithmes en considérant les aspects géométriques. Ainsi, quand une solution se situe "près du milieu" du polyèdre défini par les contraintes, il semble naturel d'orienter la recherche dans la direction de la plus forte pente. D'un autre coté, lorsqu'un point est proche des frontières, une opération de géométrie est employée pour donner l'illusion que le point est proche du centre du polyèdre. La notion "près du milieu" du polyèdre est toutefois vague et Karmarkar a levé l'ambiguïté en utilisant une représentation spéciale des contraintes. Dans l'espace transformé, le domaine réalisable est l'intersection du simplexe unité avec un cône issu de l'origine. Le centre est donc $\mathbf{1}/n$. Il est intéressant de rappeler que Dantzig a mis au point son algorithme en considérant le domaine réalisable comme l'intersection d'une région avec un simplexe, d'où le nom méthode du simplexe. Enfin, dans le cas des transformations affines l'illusion crée est que le point réalisable est à exactement une unité des frontières.

Nous avons aussi présenté des algorithmes qui emploient la technique classique des fonctions de barrière pour traiter les contraintes d'inégalité. Un algorithme naïf consiste à minimiser pour différentes valeurs décroissantes d'un paramètre positif la fonction de barrière. La suite de points converge vers la solution optimale, c'est la trajectoire centrale, et il a été établi qu'un choix particulier du paramètre donne lieu à l'algorithme de Karmarkar. La convergence est toutefois très lente et les algorithmes de la seconde catégorie reposent sur une définition de la distance d'un point avec la trajectoire centrale.

A un plus haut niveau, la division en deux catégories des algorithmes de points intérieurs reflète les deux points de vue possibles pour l'analyse des programmes linéaires. D'une part, il y a les aspects géométriques (polyèdre, point extrêmes, faces, simplexe, ...) et de l'autre coté il y a les notions avancées de l'analyse et du calcul mathématique (courbes algébriques, équations différentielles, ...). Jusqu'à maintenant, l'enseignement des techniques d'optimisation en recherche opérationnelle trace une démarcation claire entre l'étude des programmes linéaires et l'étude des programmes non linéaires; seul les aspects géométriques des programmes linéaires étant considérés. Les récentes découvertes suggèrent toutefois une approche unifiée. Ainsi, un retour aux sources s'impose! En plus des classiques Fiacco et McCormick (1968) et Huard (méthode des centres) il faut maintenant lire Bayer et Lagarias (1987a,1987b), Meggido (1989) et Vial (1989). Dans ce travail nous avons abordé les deux aspects. Cependant, cela n'a pas été couvert exhaustivement, mais à tout le moins, nous avons soulevé les points importants. Entre autre, nous avons trouvé dans la littérature une méthode générale oeuvrant dans un espace de projection pour laquelle la procédure de Karmarkar est un cas particulier. Cela a donné une première motivation pour l'utilisation de la fonction potentielle. Dans le cas des méthodes de trajectoire, nous avons énoncés les concepts mathématiques sous-jacents et nous avons constaté le lien étroit entre les fonctions de barrière et la fonction potentielle.

Nous remarquons aussi que le point de vue informatique est très souvent négligé. Ici, il a deux points importants. Le premier concerne la complexité théorique et la précision des calculs qui dépendent du nombre de bits nécessaires pour emmagasiner les données. A cet égard, la section deux résume les résultats les plus importants. D'ailleurs, nous avons constaté avec regret, que la plupart des auteurs en font un traitement très grossier. Le deuxième point concerne la mise au point des programmes informatiques. Cela n'a pas été discuté dans ce travail et nous indiquons maintenant les questions importantes.

Quelque soit l'algorithme choisi, l'effort numérique à chaque itération consiste à résoudre un système d'équations linéaires ayant la forme $\mathbf{A}\overline{\mathbf{D}}^2\mathbf{A}^T\mathbf{u} = \mathbf{r}$ où $\mathbf{D}$ est une matrice diagonale qui change à chaque itération. Le vecteur $\mathbf{r}$ est le membre de droite et le vecteur des inconnues est $\mathbf{u}$. Il y a deux approches pour solutionner de tels équations: les méthodes directes (par exemple la méthode d'élimination de Gauss, factorisation LU et factorisation de Cholesky) et les méthodes itératives (par exemple la méthode de Gauss-Siedel). Noter que la matrice des coefficients est définie positive, elle est non singulière et elle est symétrique. Dans ce cas, l'avantage d'avoir recours à la méthode de Cholesky est que la position des zéros dans la matrice de factorisation ne dépend pas de $\mathbf{D}$. Ainsi, un réarrangement des lignes de $\mathbf{A}$ est effectué au début de façon à maximiser le nombre de zéros dans la matrice de factorisation. En général, la matrice des contrainte est très creuse et la factorisation symbolique consiste à préserver au mieux cette propriété. Ceci est un problème très difficile et plusieurs heuristiques ont été étudiées. Cet aspect de la mise en oeuvre des méthodes de point intérieur et l'exploitation des structures particulières de la matrice des contraintes sont des avenues de recherche nouvelles à explorer.

En pratique les programmes informatiques les plus efficaces sont des implantations de l'algorithme dual affine (voir Adler et al. 1989; Marsten et al. 1989) et de l'algorithme de trajectoire primal-dual (voir McShane et al. 1989; Lustig et al. 1990a,b,c). Gill et al. (1986) décrit aussi une méthode de barrière mais qui ne suit pas la trajectoire centrale. Aucun de ces programmes n'est cinquante fois plus rapide que les meilleurs implantations de la méthode du simplexe! Ce qui a été observé est que les méthodes de points intérieurs ont l'avantage quand la taille des instances augmente. En fait, le nombre d'itérations requis varie entre vingt et cinquante et il augmente très lentement avec le nombre de variables. Le désavantage provient du temps requis à chaque itération pour calculer la direction de recherche.

Les expérimentations numériques menées jusqu'à maintenant conduisent à la conclusion que les algorithmes de points intérieurs ne sont pas toujours les plus rapides. Les meilleures performances, en faisant la comparaison avec la méthode du simplexe, ont été obtenues en solutionnant les problèmes d'investissement et les problèmes de détermination d'un flot à coût minimum avec des contraintes supplémentaires. Il est donc peu probable que l'algorithme de Karmarkar et l'une ou l'autre des méthodes de points intérieurs déloge l'algorithme du simplexe pour la résolution des programmes linéaires. Ce sont de nouveaux outils qui s'ajoutent à l'arsenal des praticiens de la recherche opérationnelle. Par exemple, les méthodes de points intérieurs se prêtent difficilement à des analyses de post optimisation. Ainsi, nous pouvons envisager un algorithme mixte qui effectue un certain nombre de pas dans l'intérieur du domaine réalisable et termine par quelques itérations de la méthode du simplexe. Des recherches visant à associer les classes de problèmes à la meilleure

approche de résolution restent donc à faire. Cela est relié à la "marge de manoeuvre" dans l'intérieur du domaine réalisable. Une meilleure compréhension, pas seulement géométrique mais aussi analytique, est essentielle dans le but mettre en oeuvre de meilleures méthodes.

L'avantage des méthodes de trajectoire repose sur les extensions possibles pour le traitement efficace des programmes ayant une fonction objectif non linéaire mais convexe et des contraintes linéaires. C'est la programmation monotropique et plusieurs problèmes de transport réalistes appartiennent à cette classe. Cela appelle la question des spécialisations d'une approche de point intérieur pour les problèmes ayant une structure de réseau. A cet égard, des études comparatives entre les algorithmes existants (par exemple, les méthodes de relaxation de Bertsekas) et les nouvelles méthodes restent à faire. Des approches combinées sont aussi envisageables et cela peut aussi donner lieu à de nouveaux algorithmes pour solutionner les programmes ayant des variables restreintes à des valeurs entières (design de réseau).

BIBLIOGRAPHIE

Adler I., Karmarkar N., Resende M.G.C., Veiga G., 1989, *Data Structures and Programming Techniques for the Implementation of Karmarkar's Algorithm*, **ORSA Journal on Computing**, 1(2), 84-106.

Anstreicher K.M., 1986, *A Monotonic Projective Algorithm for Fractional Linear Programming*, **Algorithmica**, 1, 483-498.

Anstreicher K.M., 1988, *Linear Programming and the Newton Barrier Flow*, **Mathematical Programming**, 41, 367-373.

Barnes E.R., 1986, *A Variation on Karmarkar's Algorithm for Solving Linear Programming Problems*, **Mathematical Programming**, 36, 174-182.

Bayer D., Lagarias J., 1987a, *The Nonlinear Geometry of Linear Programming: I. Affine and Projective Scaling Trajectories*, **Transactions of the American Mathematical Society 1989**, 314, 499-524.

Bayer D., Lagarias J., 1987b, *The Nonlinear Geometry of Linear Programming: II. Legendre Transform Coordinates and Central Trajectories*, **Transactions of the American Mathematical Society 1989**, 314, 525-548.

Bazaraa M., Jarvis J., Sherali H., 1990, **Linear Programming and Network Flows**, John Wiley & Sons.

Bazaraa M., Shetty C., 1979, **Nonlinear Programming**, John Wiley & Sons.

Blair C.E., 1986, *The Iterative Step in the Linear Programming Algorithm of N. Karmarkar*, **Algorithmica**, 1, 537-539.

Carpenter T., Lustig I., Mulvey J., Shanno D. 1990, *A Primal-Dual Interior Point Method for Convex Separable Nonlinear Programs*, Technical Report SOR 90-02, Departement of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

Dantzig G.B., 1963, **Linear Programming and Extensions**, Princeton University Press, N.J.

de Ghellinck G., Vial J.-P., 1986, *A Polynomial Newton Method for Linear Programming*, **Algorithmica**, 1, 425-453.

de Ghellinck G., Vial J.-P., 1987, *An Extension of Karmarkar's Algorithm for Solving a System of Linear Homogeneous Equations on the Simplex*, **Mathematical Programming**, 39, 79-92.

Fiacco A.V., McCormick G.P., 1968, **Nonlinear Programming: Sequential Unconstrained Minimization Techniques**, John Wiley & Sons.

Goldfarb D., Todd M., 1989, *Linear Programming*, dans G.L. Nemhauser, A.H.G. Rinnooy Kan et M.J. Todd, éditeurs, **Handbooks in Operations Research and Management Science, Volume 1**, 73-170.

Gill P.E., Murray W., Saunders M.A., Tomlin J.A., Wright M.H., 1986, *On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method*, **Mathematical Programming**, 36, 183-209.

Gonzaga C.C. 1988, *Polynomial Affine Algorithms for Linear Programming*, Dept. of Systems Engineering and Computer Sciences, Université de Rio de Janeiro, Brésil.

Gonzaga C.C. 1989, *Large-Steps Path-Following Methods for Linear Linear Programming: Barrier Function Method*, Dept. of Systems Engineering and Computer Sciences, Université de Rio de Janeiro, Brésil.

Gonzaga C.C., 1989, *An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ Operations*, dans N. Megiddo, éditeur, **Progress in Mathematical Programming**, 1-28.

Gonzaga C.C. 1989, *Canonical Projection Algorithms for Linear Programming*, **Mathematical Programming**, 43, 151-173.

Hooker J.N., 1986, *Karmarkar's Linear Programming Algorithm*, **Interfaces**, 16(4), 75-90.

Karmarkar N., 1984, *A New Polynomial-Time Algorithm for Linear Programming*, **Combinatorica**, 4(4), 373-395.

Kojima M., 1986, *Determining Basic Variables of Optimal Solutions in Karmarkar's New LP Algorithm*, **Algorithmica**, 1, 499-515.

Kojima M., Mizuno S., Yoshise A., 1989, *A Primal-Dual Interior Point Algorithm for Linear Programming*, dans N. Megiddo, éditeur, **Progress in Mathematical Programming**, 29-48.

Kortanek K., Jishan Z., 1988, *New Purification Algorithms for Linear Programming*, **Naval Research Logistics**, 35(4), 571-584.

Lustig I., Marsten R., Shanno D. 1990a, *On Implementing Mehrotra's Predictor-Corrector Interior Point Method for Linear Programming*, Technical Report SOR 90-03, Departement of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

Lustig I., Marsten R., Shanno D. 1990b, *Starting and Restarting the Primal-Dual Interior Point Method*, Technical Report SOR 90-14, Departement of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

Lustig I., Marsten R., Shanno D. 1990c, *Computational Experience with a Primal-Dual Interior Point Method for Linear Programming*, Departement of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

Marsten R., Saltzman M., Shanno D., Pierce G., Ballintijn J., 1989, *Implementation of a Dual Affine Interior Point Algorithm for Linear Programming*, **ORSA Journal on Computing**, 1(4), 287-297.

Marsten R., Subramanian R., Saltzman M., Lustig I., Shanno D., 1990, *Interior Point Methods for Linear Programming: Just Call Newton, Lagrange, and Fiacco and McCormick!*, **Interfaces**, 20(4), 105-116.

McShane K. A., Monma C.L., Shanno D., 1989, *An Implementation of a Primal-Dual Interior Point Method for Linear Programming*, **ORSA Journal on Computing**, 1(2), 70-83.

Megiddo N., 1986, *Introduction: New Approaches for Linear Programming*, **Algorithmica**, 1, 387-394.

Megiddo N., 1989, *Pathways to the Optimal Set in Linear Programming*, dans N. Megiddo, éditeur, **Progress in Mathematical Programming**, 131-158.

Monteiro R.D.C., Adler I., 1989, *Interior Path Following Primal-Dual Algorithms. Part I: Linear Programming*, **Mathematical Programming**, 44, 27-41.

Monteiro R.D.C., Adler I., 1989, *Interior Path Following Primal-Dual Algorithms. Part II: Convex Quadratic Programming*, **Mathematical Programming**, 44, 43-66.

Monteiro R.D.C., Adler I., 1990, *An Extension of Karmarkar Type Algorithm to a Class of Convex Separable Programming Problems with Global Linear Rate of Convergence*, **Mathematics of Operations Research**, 15(3), 408-422.

Papadimitriou C.H., Steiglitz K., 1982, **Combinatorial Optimization: Algorithms and Complexity**, Prentice Hall.

Renegar J., 1988, *A Polynomial-Time Algorithm, Based on Newton's Method, for Linear Programming*, **Mathematical Programming**, 40, 59-93.

Rockett A., Stevenson J., 1987, *Karmarkar's Algorithm*, **BYTE**, McGraw-Hill Publication, 146-160.

Sherali H., Skarpness B., Kim B., *An Assumption-Free Convergence Analysis for a Perturbation of the Scaling Algorithm for Linear Programs, with Application to the* L_1 *Estimation Problem*, **Naval Research Logistics**, 35(5), 473-492.

Todd M., 1988, *Exploiting Special Structure in Karmarkar's Linear Programming Algorithm*, **Mathematical Programming**, 41, 97-113.

Todd M., 1988, *Karmarkar as Dantzig-Wolfe*, Technical Report 782, School of Operations Research and Industrial Engineering, College of Engineering, Cornell University.

Todd M., Burrell P., 1986, *An Extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables*, **Algorithmica**, 1, 409-424.

Todd M., Ye Y., 1990, *A Centered Projective Algorithm for Linear Programming*, **Mathematics of Operations Research**, 15(3), 508-529.

Tomlin J., 1989, *A Note on Comparing Simplex and Interior Methods for Linear Programming* dans **Lecture Notes in Mathematics 1405**, Springer Verlag, 191-220

Vaidya P., 1989, *A Locally Well-Behaved Potential Function and a Simple Newton-Type Method for Finding the Center of a Polytope*, dans N. Megiddo, éditeur, **Progress in Mathematical Programming**, 79-90.

Vanderbei R.J., Meketon M.S., Freedman B.A., 1986, *A Modification of Karmarkar's Linear Programming Algorithm*, **Algorithmica**, 1, 395-407.

Vial J.-P., 1989, *A Unified Approach to Projective Algorithms for Linear Programming* dans **Lecture Notes in Mathematics 1405**, Springer Verlag, 191-220

Ye Y., 1988, *A Class of Potential Functions for Linear Programming*, Manuscript, Department of Management Sciences, The University of Iowa, Iowa City, Iowa.